DL850/DL850V
ScopeCorder
Communication Interface
USER'S MANUAL

YOKOGAWA
Yokogawa Meters & Instruments Corporation

IM DL850-17EN
2nd Edition

Thank you for purchasing the DL850 ScopeCorder or DL850V ScopeCorder Vehicle Edition (hereinafter, “DL850/DL850V” will refer to both of these products).
This Communication Interface User’s Manual explains the following interface features and commands.

- Ethernet interface
- USB interface
- GP-IB interface (optional)

To ensure correct use, please read this manual thoroughly before operation.
Keep this manual in a safe place for quick reference in the event a question arises. The following manuals, including this one, are provided as manuals for the DL850/DL850V.
Please read all manuals.

Manual Title	Manual No.	Description
DL850/DL850V ScopeCorder Features Guide	IM DL850-01EN	The supplied CD contains the PDF file of this manual. This manual explains all the DL850/DL850V features other than the communication interface features.
DL850/DL850V ScopeCorder User’s Manual	IM DL850-02EN	The supplied CD contains the PDF file of this manual. The manual explains how to operate the DL850/DL850V.
DL850/DL850V ScopeCorder Getting Started Guide	IM DL850-03EN	The manual explains the handling precautions and basic operations of the DL850/DL850V and provides an overview of its features.
DL850/DL850V ScopeCorder Communication Interface User’s Manual	IM DL850-17EN	This manual. The supplied CD contains the PDF file of this manual. The manual explains the DL850/DL850V communication interface features and instructions on how to use them.

Notes

- The contents of this manual are subject to change without prior notice as a result of continuing improvements to the instrument’s performance and functions. The figures given in this manual may differ from those that actually appear on your screen.
- Every effort has been made in the preparation of this manual to ensure the accuracy of its contents. However, should you have any questions or find any errors, please contact your nearest YOKOGAWA dealer.
- Copying or reproducing all or any part of the contents of this manual without the permission of Yokogawa Electric Corporation is strictly prohibited.
- The TCP/IP software of this product and the documents concerning it have been developed/created by YOKOGAWA based on the BSD Networking Software, Release 1 that has been licensed from the Regents of the University of California.

Trademarks

- Microsoft, Internet Explorer, MS-DOS, Windows, Windows NT, and Windows XP are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
- Adobe, Acrobat, and PostScript are trademarks of Adobe Systems Incorporated.
- GIGAZoom ENGINE is a registered trademark of YOKOGAWA Electric Corporation.
- In this manual, the TM and ® symbols do not accompany their respective trademark or registered trademark names.
- Other company and product names are trademarks or registered trademarks of their respective companies.

Revisions

- 1st Edition: June 2010
- 2nd Edition: September 2010

2nd Edition: September 2010 (YK)
All Rights Reserved, Copyright © 2010 Yokogawa Meters & Instruments Corporation

About the USB Interface and Ethernet Interface

- To use the USB communication features, your PC must have the following:
  - DL850/DL850V library (TMCTL)
  - USB device driver for connecting the DL850/DL850V to the PC
- To use the Ethernet communication features, your PC must have the following:
  - DL850/DL850V library (TMCTL)

  To download the library and driver listed above, go to the following Web page, and then browse to the download page.
  http://www.yokogawa.co.jp/tm/F-SOFT/

How to Use This Manual

Structure of the Manual

This manual contains six chapters and an appendix.

Chapter 1 Ethernet Interface

Describes the features and specifications of the Ethernet interface.

Chapter 2 USB Interface

Describes the features and specifications of the USB interface.

Chapter 3 GP-IB Interface (Optional)

Describes the features and specifications of the GP-IB interface.

Chapter 4 Programming Overview

Describes command syntax and other programming information.

Chapter 5 Commands

Describes every command individually.

Chapter 6 Status Reports

Describes the status byte, various registers, and queues.

Appendix

Provides reference material such as an ASCII character code table.

Index

Conventions Used in This Manual

Notes and Cautions

The notes and cautions in this manual are categorized using the following symbols.

WARNING Calls attention to actions or conditions that could cause serious or fatal injury to the user, and precautions that can be taken to prevent such occurrences.

CAUTION Calls attentions to actions or conditions that could cause light injury to the user or damage to the instrument or user's data, and precautions that can be taken to prevent such occurrences.

Note Calls attention to information that is important for proper operation of the instrument.

Character Notations

Hard Key Names and Soft Key Names in Bold Characters
Indicate panel keys that are used in the procedure and soft keys and menu items that appear on the screen.

SHIFT+Panel Key
When SHIFT+panel key appears in a procedural explanation, it means to press the shift key so that its indicator lights, and then to press the indicated panel key. A setup menu for the item written in purple above the key that you pressed appears on the screen.

Unit

k	Denotes 1000. Example:100 kS/s (sample rate)
K	Denotes 1024. Example:720 KB (file size)

Metasyntax

The following table contains the symbols that are used in the syntax discussed mainly in chapters 4 and 5. These symbols are referred to as BNF (Backus-Naur Form) symbols. For details on how to write data using these symbols, see pages 4-6 and 4-7.

Symbol	Description	Syntax Example	Example
<>	A defined value	CHANnel<x> <x> = 1 to 4	CHANNEL2
{}	Select an option in { }	COUPling {AC|DC|DC50|GND}	COUPLING AC
|	Exclusive OR		
[]	Can be omitted	TRIGger [:SIMPle]:SLOPe	TRIGger:SLOPe

Contents

How to Use This Manual iii

Chapter 1 Ethernet Interface

1.1 Component Names and Functions 1-1
1.2 Ethernet Interface Features and Specifications 1-2
1.3 Connecting to the Ethernet Interface 1-4
1.4 Configuring the DL850/DL850V Ethernet Settings 1-5

Chapter 2 USB Interface

2.1 Component Names and Functions 2-1
2.2 USB Interface Features and Specifications 2-2
2.3 Connecting to the USB Interface 2-3
2.4 Configuring the DL850/DL850V USB Settings 2-4

Chapter 3 GP-IB Interface (Optional)

3.1 Component Names and Functions 3-1
3.2 Connecting GP-IB Cables (Optional) 3-2
3.3 GP-IB Interface Features 3-4
3.4 GP-IB Interface Specifications 3-5
3.5 Configuring the DL850/DL850V GP-IB Settings 3-6
3.6 Responses to Interface Messages 3-7

Chapter 4 Programming Overview

4.1 Messages 4-1
4.2 Commands 4-3
4.3 Response 4-5
4.4 Data 4-6
4.5 Synchronization with the Controller 4-8

Chapter 5 Commands

5.1 List of Commands 5-1
5.2 ACQuire Group 5-25
5.3 ASETup Group 5-27
5.4 CALibrate Group 5-28
5.5 CAPTure Group 5-29
5.6 CHANnel Group 5-32
5.7 CLEar Group 5-55
5.8 COMMunicate Group 5-56
5.9 CURSor Group 5-58
5.10 DISPlay Group 5-70
5.11 EVENt Group 5-73
5.12 FFT Group 5-74
5.13 FILE Group 5-77
5.14 GONogo Group 5-80
5.15 HCOPy Group 5-85
5.16 HISTory Group 5-86
5.17 IMAGe Group 5-89
5.18 INITialize Group 5-90

5.19 LSTart Group 5-91
5.20 MATH Group 5-92
5.21 MEASure Group 5-97
5.22 MONitor Group 5-104
5.23 MTRigger Group 5-106
5.24 RECall Group 5-107
5.25 SEARch Group 5-108
5.26 SNAP Group 5-111
5.27 SSTart Group 5-112
5.28 STARt Group 5-113
5.29 STATus Group 5-114
5.30 STOP Group 5-115
5.31 STORe Group 5-116
5.32 SYSTem Group 5-117
5.33 TIMebase Group 5-121
5.34 TRIGger Group 5-122
5.35 WAVeform Group 5-133
5.36 XY Group 5-137
5.37 ZOOM Group 5-139
5.38 Common Command Group 5-141

Chapter 6 Status Reports

6.1 About Status Reports 6-1
6.2 Status Byte 6-3
6.3 Standard Event Register 6-4
6.4 Extended Event Register 6-5
6.5 Output and Error Queues 6-6

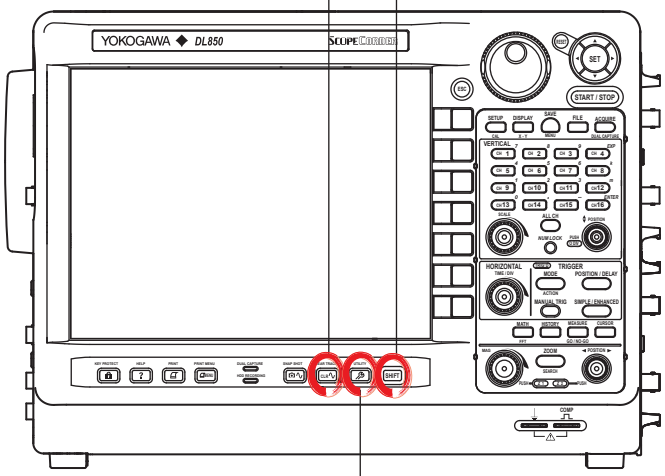
Appendix

Index

1.1 Component Names and Functions

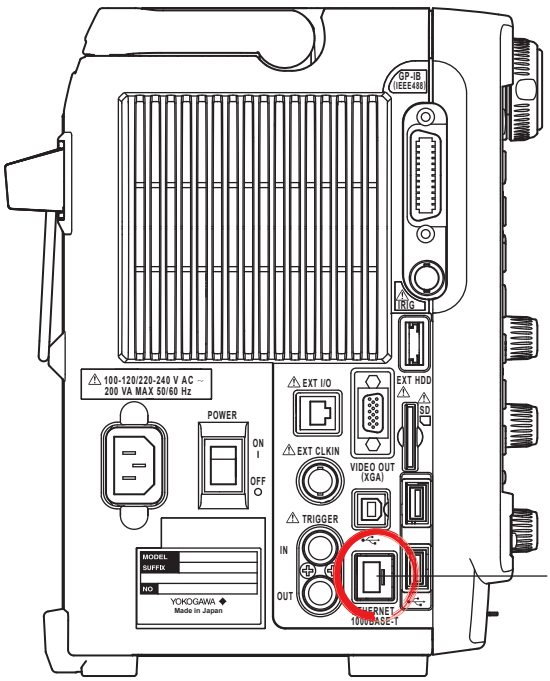
Front Panel

SHIFT key and CLEAR TRACE key
Press these keys to switch from remote mode, in which settings and operations are performed through remote commands, to local mode, in which operations can be performed using the DL850/DL850V keys. These keys are disabled when local lockout (see page 1-2) has been activated by a controller.



UTILITY key (page 1-5)
Press this key to set the communication interface and timeout value for remote control and to set the user name and password for user authentication.

Side Panel



Ethernet port
This port is for connecting the DL850/DL850V to a controller (such as a PC) using an Ethernet cable. For details on how to connect the DL850/DL850V to a controller, see page 1-4.

1.2 Ethernet Interface Features and Specifications

Ethernet Interface Features

Reception Features

Allows you to specify the same settings that you can using the front panel keys.

Receives output requests for measured and computed data, panel setting data, and error codes.

Transmission Features

The DL850/DL850V can transmit measured and computed data.

The DL850/DL850V can transmit panel setting data and the status byte.

The DL850/DL850V can transmit error codes when errors occur.

Ethernet Interface Specifications

Number of ports:	1
Electrical and mechanical specifications:	IEEE802.3
Data rate:	100 Mbps max.
Communication protocol:	TCP/IP
Connector:	RJ-45
Port number:	10001/tcp

Data Transfer Rate

The following table contains approximations of how much time it takes for the DL850/DL850V to transmit waveform data.

Model:	DL850/DL850V
Controller:	PC: Pentium4 3.2 GHz, OS: Windows XP
Network adapter:	Intel PRO/1000 GT Desktop Adapter
Programming language:	Visual C++

Number of Data Points	Byte Data	Word Data	ASCII Data
1000	Approx. 1 ms	Approx. 1 ms	Approx. 30 ms
10000	Approx. 1 ms	Approx. 2 ms	Approx. 300 ms
100000	Approx. 10 ms	Approx. 11 ms	Approx. 3 s
1000000	Approx. 100 ms	Approx. 125 ms	Approx. 30 s

Switching between Remote and Local Modes

Switching from Local to Remote Mode

The DL850/DL850V switches to remote mode when it is in local mode and it receives a `:COMMunicate:REMote ON` command from the PC.

- "REMOTE" appears at the top center of the screen once the DL850/DL850V is in remote mode.
- All keys except the **SHIFT+CLEAR TRACE** keys are disabled.
- The local mode settings are retained even when the DL850/DL850V switches to remote mode.

Switching from Remote to Local Mode

When the DL850/DL850V is in Remote mode and you press **SHIFT+CLEAR TRACE**, the DL850/DL850V switches to local mode. However, this does not work if the DL850/DL850V has received a `:COMMunicate:LOCKout ON` command from the PC. The DL850/DL850V switches to local mode when it receives a `:COMMunicate:REMote OFF` command from the PC, regardless of the local lockout state.

- The "REMOTE" indicator at the top center of the screen disappears once the DL850/DL850V is in local mode.
- All keys are enabled.
- The settings in remote mode are retained even when the DL850/DL850V switches to local mode.

Note

You cannot use the Ethernet interface at the same time as other interfaces (GP-IB and USB interfaces).

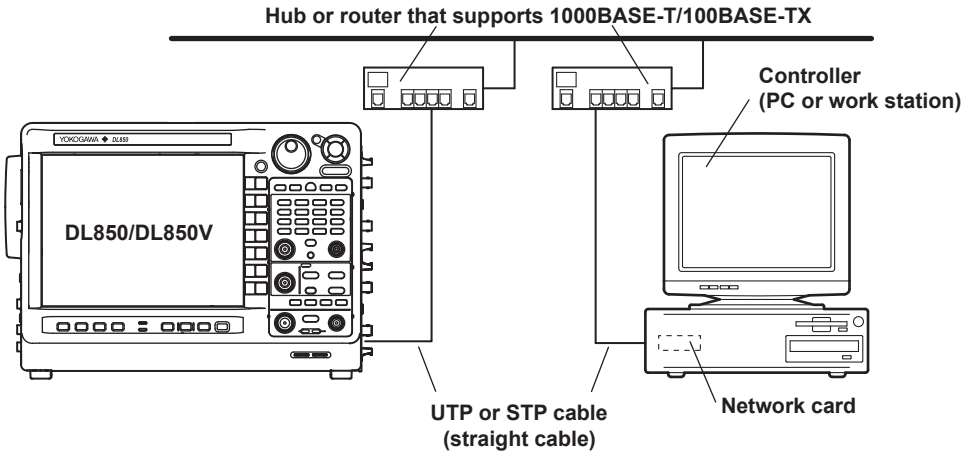
User Verification Function

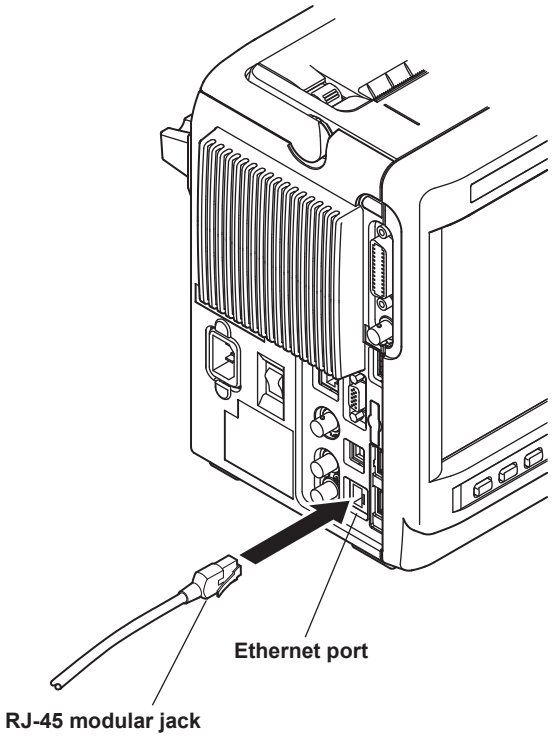
To connect the DL850/DL850V to a network as an FTP server, you have to enter a user name and password in the Ethernet-communication-interface settings. To set the user name and password, access the UTILITY menu, the Network menu, and then the FTP Server screen. For details, see section 17.3, “Accessing the DL850/DL850V from a PC (FTP Server)” in the *DL850/DL850V User’s Manual*, IM DL850-02EN.

1.3 Connecting to the Ethernet Interface

Connection Procedure

Connect a UTP (Unshielded Twisted-Pair) or STP (Shielded Twisted-Pair) cable that is connected to a hub or other network device to the Ethernet port on the DL850/DL850V side panel.





Precautions to Be Taken When Connecting Probes

- To connect the DL850/DL850V to a PC, be sure to use straight cables and to connect through a hub or router. Proper operation is not guaranteed for a one-to-one connection using a cross cable.
- Use a network cable that conforms to the transfer speed of your network.

Note

For details on how to connect the DL850/DL850V to a network, see section 17.1, “Connecting the DL850/DL850V to a Network” in the *DL850/DL850V User’s Manual*, IM DL850-02EN.

1.4 Configuring the DL850/DL850V Ethernet Settings

This section explains the settings listed below. You must configure these settings when controlling the DL850/DL850V remotely through an Ethernet interface.

- Communication interface
- Network connection timeout setting

UTILITY Remote Ctrl Menu

Press **UTILITY** and then the **Remote Ctrl** soft key to display the following menu.

Set Device to Network.

Set the timeout value (using the jog shuttle).

Note

Only use the selected communication interface. If you send commands simultaneously from another communication interface that has not been selected, the DL850/DL850V will not execute the commands properly.

Configuring TCP/IP Settings

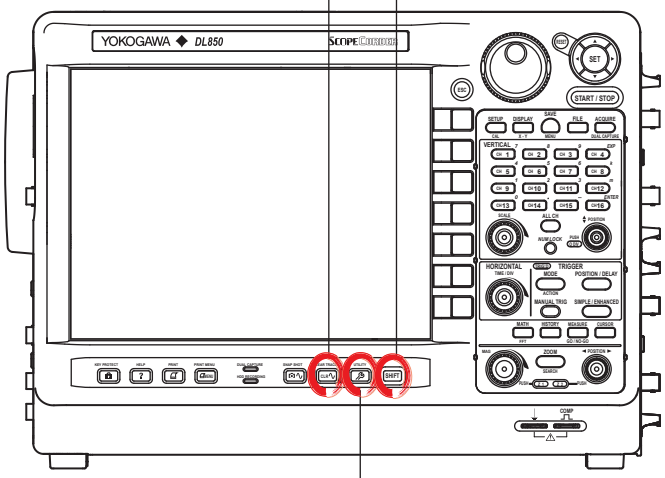
To use the Ethernet interface features, you must specify the following TCP/IP settings.

- IP address
- Subnet mask
- Default gateway

For details on how to specify these settings, see section 17.2, “Configuring TCP/IP Settings” in the *DL850/DL850V User’s Manual*, IM DL850-02EN.

2.1 Component Names and Functions

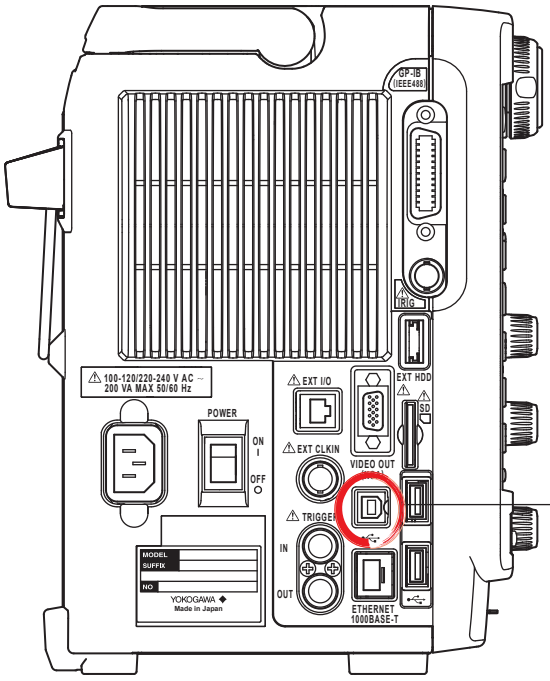
Front Panel



SHIFT key and CLEAR TRACE key
Press these keys to switch from remote mode, in which settings and operations are performed through remote commands, to local mode, in which operations can be performed using the DL850/DL850V keys. These keys are disabled when local lockout (see page 2-2) has been activated by a controller.

UTILITY key (page 2-4)
Press this key to set the communication interface for remote control and to enable remote control through the USB ports (when you want to use remote commands).

Side Panel



USB ports
These ports are for connecting the DL850/DL850V to a controller (such as a PC) using a USB cable. For details on how to connect the DL850/DL850V to a controller, see page 2-3.

2.2 USB Interface Features and Specifications

USB Interface Features

Reception Features

Allows you to specify the same settings that you can using the front panel keys.
Receives output requests for measured and computed data, panel setting data, and error codes.

Transmission Features

The DL850/DL850V can transmit measured and computed data.
The DL850/DL850V can transmit panel setting data and the status byte.
The DL850/DL850V can transmit error codes when errors occur.

USB Interface Specifications

Electrical and mechanical specifications: USB 2.0
Connector: Type B connector (receptacle)
Number of ports: 2
Power supply: Self powered
System requirements: A PC with a USB port, running Windows 7 (32 bit), Windows Vista (32 bit), or Windows XP (32 bit, SP2 or later). A separate device driver is required to enable the connection with the PC.

Data Transfer Rate

The following table contains approximations of how much time it takes for the DL850/DL850V to transmit waveform data.

Model: DL850/DL850V
Controller: PC: Pentium 4 3.2 GHz, USB 2.0 (ICH6), OS: Windows XP
Programming language: Visual C++

Number of Data Points	Byte Data	Word Data	ASCII Data
1000	Approx. 1 ms	Approx. 1 ms	Approx. 30 ms
10000	Approx. 1 ms	Approx. 2 ms	Approx. 300 ms
100000	Approx. 16 ms	Approx. 15 ms	Approx. 3 s
1000000	Approx. 111 ms	Approx. 170 ms	Approx. 30 s

Switching between Remote and Local Modes

Switching from Local to Remote Mode

The DL850/DL850V switches to remote mode when it is in local mode and it receives a `:COMMunicate:REMote ON` command from the PC.

- "REMOTE" appears at the top center of the screen once the DL850/DL850V is in remote mode.
- All keys except the **SHIFT+CLEAR TRACE** keys are disabled.
- The local mode settings are retained even when the DL850/DL850V switches to remote mode.

Switching from Remote to Local Mode

When the DL850/DL850V is in Remote mode and you press **SHIFT+CLEAR TRACE**, the DL850/DL850V switches to local mode. However, this does not work if the DL850/DL850V has received a `:COMMunicate:LOCKout ON` command from the PC. The DL850/DL850V switches to local mode when it receives a `:COMMunicate:REMote OFF` command from the PC, regardless of the local lockout state.

- The "REMOTE" indicator at the top center of the screen disappears once the DL850/DL850V is in local mode.
- All keys are enabled.
- The settings in remote mode are retained even when the DL850/DL850V switches to local mode.

Note

You cannot use the USB interface at the same time as other interfaces (GP-IB and Ethernet interfaces).

2.3 Connecting to the USB Interface

Precautions to Be Taken When Connecting Probes

- Be sure to insert the USB cable connectors firmly into the USB ports.
- If you are connecting multiple devices by using a USB hub, connect the DL850/DL850V to the USB hub port that is closest to the port that the controller is connected to.
- Do not connect a USB cable (type B) to the GO/NO-GO output terminal. Doing so may damage the DL850/DL850V.
- Do not connect or remove USB cables from the time when the DL850/DL850V is turned on until operation becomes available (approximately 20 to 30 seconds). Doing so may damage the DL850/DL850V.

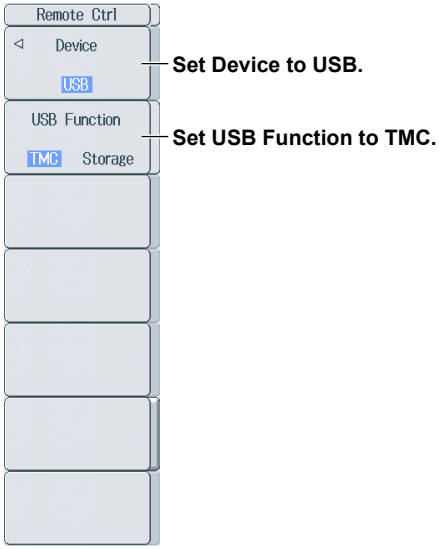
2.4 Configuring the DL850/DL850V USB Settings

This section explains the settings listed below. You must configure these settings when controlling the DL850/DL850V remotely through a USB interface.

- Communication interface

UTILITY Remote Ctrl Menu

Press **UTILITY** and then the **Remote Ctrl** soft key to display the following menu.

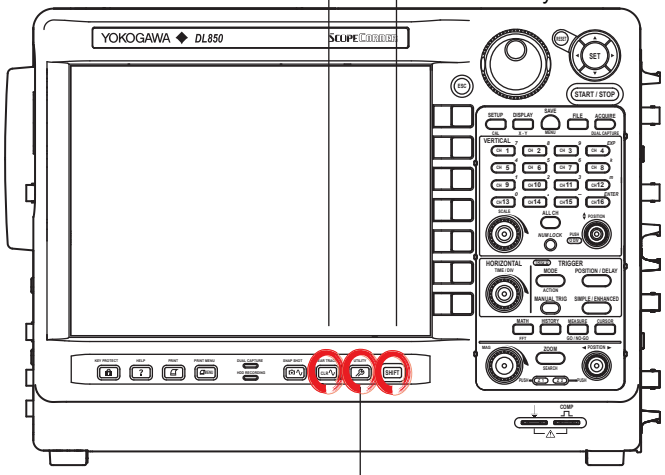


Note

- Only use the selected communication interface. If you send commands simultaneously from another communication interface that has not been selected, the DL850/DL850V will not execute the commands properly.
- To remotely control the DL850/DL850V through a USB port using communication commands, set USB Function, shown in the figure above, to TMC, and then carry out the following steps.
- To activate the USB Function settings, you need to restart the DL850/DL850V. Turn off the DL850/DL850V power switch, wait ten seconds or more, and then turn on the switch.
- Install the YOKOGAWA USB TMC (Test and Measurement Class) driver on your PC. For information about how to obtain the YOKOGAWA USB TMC driver, contact your nearest YOKOGAWA dealer. You can also access the YOKOGAWA USB driver download webpage and download the driver.
  http://www.yokogawa.co.jp/tm/F-SOFT/
- Do not use USB TMC drivers (or software) supplied by other companies.

3.1 Component Names and Functions

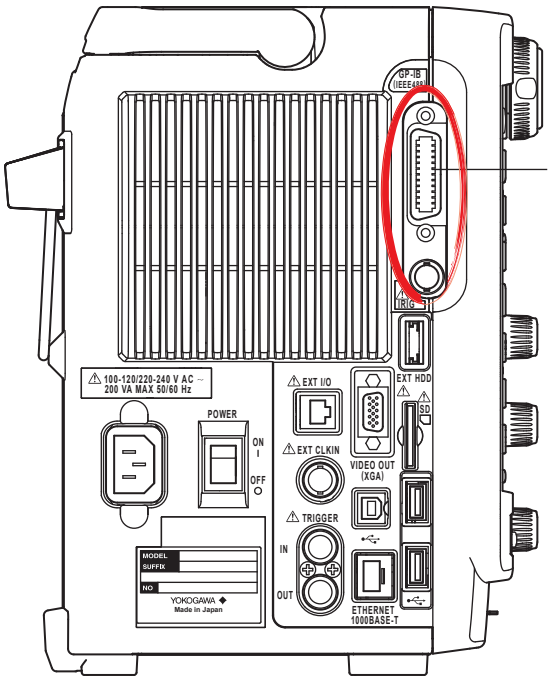
Front Panel



SHIFT key and CLEAR TRACE key
Press these keys to switch from remote mode, in which settings and operations are performed through remote commands, to local mode, in which operations can be performed using the DL850/DL850V keys. These keys are disabled when local lockout (see page 3-7) has been activated by a controller.

UTILITY key (page 3-5)
Press this key to set the communication interface for remote control and the GP-IB address.

Side Panel



GP-IB port
This port is for connecting the DL850/DL850V to a controller (such as a PC) using a GP-IB cable. For details on how to connect the DL850/DL850V to a controller, see page 3-2.

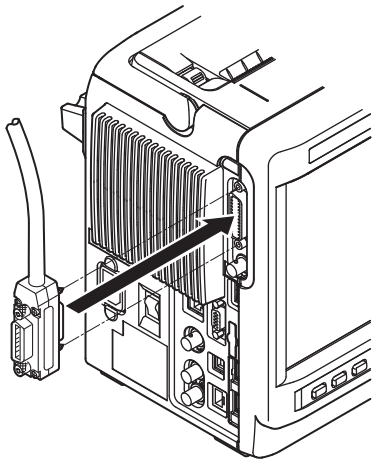
3.2 Connecting GP-IB Cables (Optional)

GP-IB Cable

The DL850/DL850V is equipped with an IEEE St'd 488-1978 24-pin GP-IB connector. Use GP-IB cables that comply with IEEE St'd 488-1978.

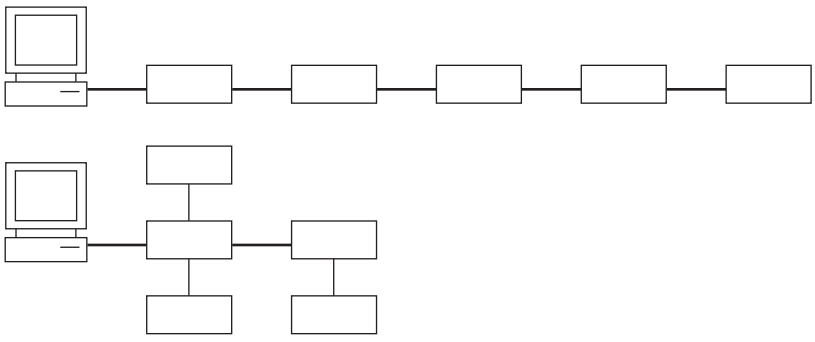
Connection Procedure

Connect a GP-IB cable as shown below.



Precautions to Be Taken When Connecting Probes

- Securely fasten the GP-IB cable connector screws.
- On the PC end, use a GP-IB board (or card) made by National Instruments. For more details, see section 3.4.
- The DL850/DL850V may not operate properly if the DL850/DL850V is connected to the PC through converters (such as a GP-IB to USB converter). For more details, contact your nearest YOKOGAWA dealer.
- Several cables can be used to connect multiple devices. However, no more than 15 devices, including the controller, can be connected on a single bus.
- When connecting multiple devices, you must assign a unique address to each device.
- Use cables that are 2 m or shorter in length to connect devices.
- Keep the total length of the cables under 20 m.
- When devices are communicating, have at least two-thirds of the devices on the bus turned on.
- To connect multiple devices, use a star or daisy-chain configuration as shown below. Loop and parallel configurations are not allowed.



CAUTION

Be sure to turn off the PC and the DL850/DL850V when connecting or removing communication cables. Otherwise, erroneous operation may result, or the internal circuitry may break.

3.3 GP-IB Interface Features

GP-IB Interface Features

Listener Capabilities

- Allows you to specify the same DL850/DL850V settings that you can using the front panel keys. You cannot turn the power on and off or change communication settings.
- Receives output requests for measured and computed data, panel setting data, and error codes.
- Receives status report commands and other commands.

Talker Capabilities

The DL850/DL850V can transmit measured and computed data.
The DL850/DL850V can transmit panel setting data and the status byte.
The DL850/DL850V can transmit error codes when errors occur.

Note

Talk-only, listen-only, and controller capabilities are not available on the DL850/DL850V.

Switching between Remote and Local Modes

Switching from Local to Remote Mode

The DL850/DL850V switches to remote mode when it is in local mode and it receives a REN (Remote Enable) message from the PC.

- "REMOTE" appears at the top center of the screen once the DL850/DL850V is in remote mode.
- All keys except the **SHIFT+CLEAR TRACE** keys are disabled.
- The local mode settings are retained even when the DL850/DL850V switches to remote mode.

Switching from Remote to Local Mode

When the DL850/DL850V is in Remote mode and you press **SHIFT+CLEAR TRACE**, the DL850/DL850V switches to local mode. These keys are disabled when local lockout (see page 3-7) has been activated by a controller.

- The "REMOTE" indicator at the top center of the screen disappears once the DL850/DL850V is in local mode.
- All keys are enabled.
- The settings in remote mode are retained even when the DL850/DL850V switches to local mode.

Note

You cannot use the GP-IB interface simultaneously with other interfaces (USB and Ethernet interfaces).

3.4 GP-IB Interface Specifications

GP-IB Interface Specifications

Electrical and mechanical specifications: IEEE St'd 488-1978
Functional specifications: See the table below.
Protocol: IEEE St'd 488.2-1992
Code: ISO (ASCII) codes
Mode: Addressable mode
Address setup: Press **UTILITY** and then the Remote Ctrl soft key. Then, set the network interface (Device) to GPIB and the address to a number from 0 to 30.
Clearing remote mode: Press **SHIFT**+**CLEAR TRACE** to switch the DL850/DL850V to local mode.
These keys are disabled when local lockout has been activated by a controller.

Functional Specifications

Function	Subset Name	Description
Source handshaking	SH1	Full source handshaking capability
Acceptor handshaking	AH1	Full acceptor handshaking capability
Talker	T6	Basic talker capability, serial polling, untalk on MLA (My Listen Address), and no talk-only capability
Listener	L4	Basic listener capability, unlisten on MTA (My Talk Address), and no listen-only capability
Service request	SR1	Full service request capability
Remote local	RL1	Full remote/local capability
Parallel polling	PP0	No parallel poll capability
Device clear	DC1	Full device clear capability
Device trigger	DT0	No device trigger capability
Controller	C0	No controller capability
Electric characteristics	E1	Open collector

Data Transfer Rate

The following table contains approximations of how much time it takes for the DL850/DL850V to transmit waveform data.

Model: DL850/DL850V
Controller: PC: Pentium 4 3.2 GHz, GP-IB (GPIB-USB-B), OS: Windows XP
Programming language: Visual C++

Number of Data Points	Byte Data	Word Data	ASCII Data
1000	Approx. 24 ms	Approx. 20 ms	Approx. 54 ms
10000	Approx. 31 ms	Approx. 44 ms	Approx. 510 ms
100000	Approx. 170 ms	Approx. 310 ms	Approx. 5 s
1000000	Approx. 1600 ms	Approx. 3100 ms	Approx. 50 s

3.5 Configuring the DL850/DL850V GP-IB Settings

This section explains the settings listed below. You must configure these settings when controlling the DL850/DL850V remotely through a GP-IB interface.

- Communication interface
- GP-IB address

UTILITY Remote Ctrl Menu

Press **UTILITY** and then the **Remote Ctrl** soft key to display the following menu.

Set Device to GPIB.

Set the address (0-30).

Note

- Only use the selected communication interface. If you send commands simultaneously from another communication interface that has not been selected, the DL850/DL850V will not execute the commands properly.
- When the controller is communicating with the DL850/DL850V or with other devices through GP-IB, do not change the address.
- Each device that is connected by GP-IB has its own unique address in the GP-IB system. This address is used to distinguish one device from other devices. Therefore, you must assign a unique address to the DL850/DL850V when connecting it to a PC or other device.

3.6 Responses to Interface Messages

Responses to Interface Messages

Responses to Uni-Line Messages

- **IFC (Interface Clear)**
  Clears the talker and listener functions. Stops data transmission if it is in progress.
- **REN (Remote Enable)**
  Switches between remote and local modes.

  IDY (Identify) is not supported.

Responses to Multi-Line Messages (Address commands)

- **GTL (Go To Local)**
  Switches to local mode.
- **SDC (Selected Device Clear)**
  - Clears the program message (command) being received and the output queue (see page 6-6).
  - Discards `*OPC` and `*OPC?` commands that are being executed.
  - Immediately aborts `*WAI` and `COMMunicate:WAIT`.

  PPC (Parallel Poll Configure), GET (Group Execute Trigger), and TCT (Take Control) are not supported.

Responses to Multi-Line Messages (Universal commands)

- **LLO (Local Lockout)**
  Disables the **CLEAR TRACE** key on the front panel to prohibit switching to the local mode.
- **DCL (Device Clear)**
  Performs the same operation as SDC.
- **SPE (Serial Poll Enable)**
  Sets the talker function on all devices on the bus to serial polling mode. The controller will poll each device in order.
- **SPD (Serial Poll Disable)**
  Clears the talker function's serial poll mode on all devices on the bus.

  PPU (Parallel Poll Unconfigure) is not supported.

What Are Interface Messages?

Interface messages are commands that a controller transmits. They are also referred to as interface commands or bus commands. They are classified as follows:

Uni-line Messages

Uni-line messages are sent over a single control line. The following three terminators are available.

- IFC (Interface Clear)
- REN (Remote Enable)
- IDY (Identify)

Multi-line Messages

Multi-line messages are sent over eight data lines. The messages are grouped as follows:

- **Address Commands**

  Some address commands are valid when a device is designated as a listener, and some are valid when it is designated as a talker. The following five commands are available.

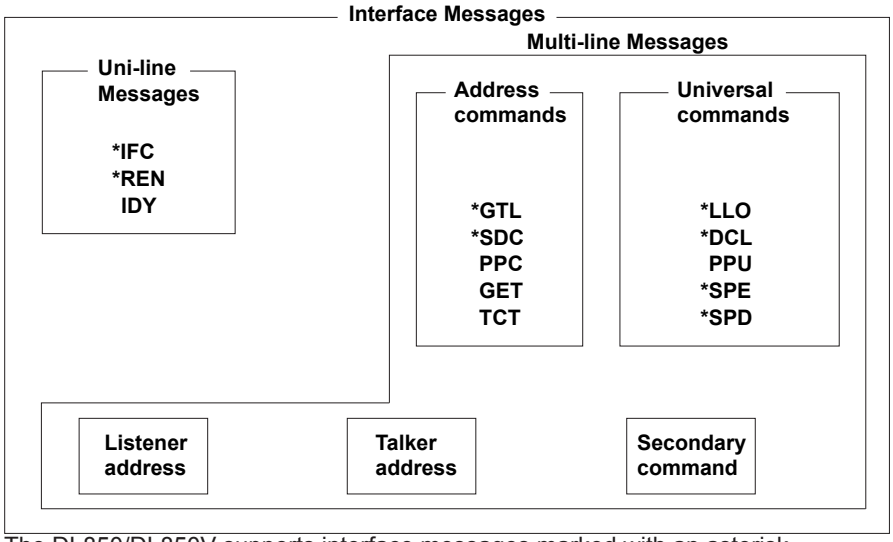
  Commands available to a device designated as a listener
  - GTL (Go To Local)
  - SDC (Selected Device Clear)
  - PPC (Parallel Poll Configure)
  - GET (Group Execute Trigger)

  Commands available to a device designated as a talker
  - TCT (Take Control)

- **Universal Commands**

  Universal commands are available to all devices regardless of their listener or talker designation. The following five commands are available.
  - LLO (Local Lockout)
  - DCL (Device Clear)
  - PPU (Parallel Poll Unconfigure)
  - SPE (Serial Poll Enable)
  - SPD (Serial Poll Disable)

There are other interface messages: listener-address, talk-address, and secondary commands.



The DL850/DL850V supports interface messages marked with an asterisk.

Note

Difference between SDC and DCL

In multi-line messages, SDC messages are those that require talker or listener designation and DCL messages are those that do not require the designation. Therefore, the SDC command affects a specific device while the DCL command affects all devices on the bus.

4.1 Messages

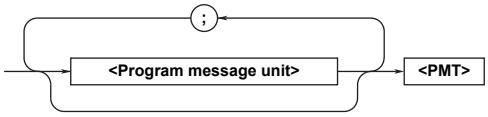
Messages

Messages are used to exchange information between the controller and the DL850/DL850V. Messages that are sent from the controller to the DL850/DL850V are called program messages, and messages that are sent from the DL850/DL850V back to the controller are called response messages.

If a program message contains a command that requests a response (a query), the DL850/DL850V returns a response message upon receiving the program message. The DL850/DL850V returns a single response message in response to a single program message.

Program Message

The program message syntax is as follows:



<Program Message Unit>

A program message consists of one or more program message units. Each unit corresponds to one command. The DL850/DL850V executes the commands in the order that they are received. Separate each program message unit with a semicolon.

For details on the program message syntax, see the next section.

Example `:ACQuire:MODE NORMal;COUNt 1<PMT>`

Unit Unit

<PMT>

<PMT> is a program message terminator. The following three terminators are available.

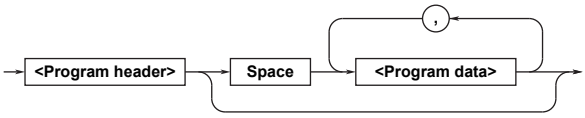
NL (new line): Same as LF (line feed). ASCII code "0AH"

`^END`: The END message as defined by IEEE 488.1 (The data byte that is sent with the END message is the last data byte of the program message.)

`NL^END`: NL with an END message attached. (NL is not included in the program message.)

Program Message Unit Syntax

The program message unit syntax is as follows:



<Program Header>

The program header indicates the command type. For details, see page 4-3.

<Program Data>

Attach program data if there are conditions that are required to execute a command. Separate the program data from the header with a space (ASCII code 20H). If there are multiple data values, separate each data value with a comma.

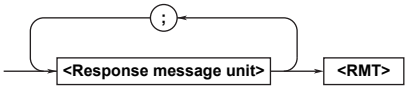
For details, see page 4-6.

Example `:ACQuire:MODE NORMal<PMT>`

Header Data

Response Message

The response message syntax is as follows:



<Response Message Unit>

A response message consists of one or more response message units; each response message unit corresponds to one response.

Separate each response message unit with a semicolon.

For details on the response message syntax, see the next page.

Example `:ACQUIRE:MODE NORMAL;COUNT 1<RMT>`

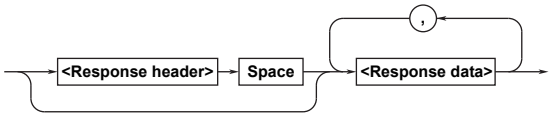
Unit Unit

<RMT>

RMT stands for "response message terminator." The response message terminator is `NL^END`.

Response Message Unit Syntax

The response message unit syntax is as follows:



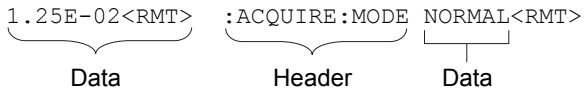
<Response Header>

A response header sometimes precedes the response data. A space separates the data from the header. For details, see page 4-5.

<Response Data>

Response data contains the content of the response. If there are multiple data values, each data value is separated by a comma. For details, see page 4-5.

Example



If there are multiple queries in a program message, responses are returned in the same order that the queries were received in. In most cases, a single query returns a single response message unit, but there are a few queries that return multiple units. The first response message unit always corresponds to the first query, but the nth response unit may not necessarily correspond to the nth query. If you want to make sure that every response is retrieved, divide the program messages into individual messages.

Precautions to Be Taken when Exchanging Messages

- If the controller sends a program message that does not contain a query, the controller can send the next program message at any time.
- If the controller sends a program message that contains a query, the controller must finish receiving the response message before it can send the next program message. If the controller sends the next program message before receiving the response message in its entirety, an error will occur. A response message that is not received in its entirety will be discarded.
- If the controller tries to receive a response message when there is none, an error will occur. If the controller tries to receive a response message before the transmission of the program message is complete, an error will occur.
- If the controller sends a program message containing multiple message units, but the message contains incomplete units, the DL850/DL850V will try to execute the ones that are believed to be complete. However, these attempts may not always be successful. In addition, if such a message contains queries, the DL850/DL850V may not necessary return responses.

Deadlock

The DL850/DL850V can store at least 1024 bytes of messages in its transmit and receive buffers (the number of available bytes varies depending on the operating conditions). If both the transmit and receive buffers become full at the same time, the DL850/DL850V will no longer be able to operate. This condition is called a deadlock. If this happens, you can resume operation after you have discarded response messages.

Deadlock will not occur if the program message (including the <PMT>) is kept below 1024 bytes. Program messages that do not contain queries never cause deadlocks.

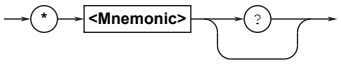
4.2 Commands

Commands

There are three types of commands (program headers) that a controller may send to the DL850/DL850V. The commands differ in their program header formats.

Common Command Header

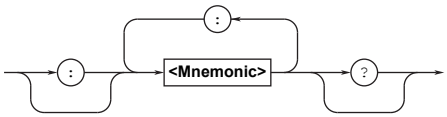
Commands that are defined in IEEE 488.2-1987 are called common commands. The header format of a common command is shown below. Be sure to include an asterisk (*) at the beginning of a common command.



Common command example `*CLS`

Compound Header

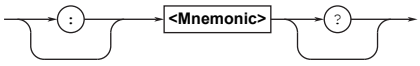
Other commands that are specific to the DL850/DL850V are classified and arranged in a hierarchy according to their functions. The compound header syntax is shown below. Be sure to use a colon to specify a lower hierarchical level.



Compound header example `:ACQuire:MODE`

Simple Header

These commands are functionally independent and are not contained within a hierarchy. The format of a simple header is shown below.



Simple header example `:STARt`

Note

A <mnemonic> is an alphanumeric character string.

When Concatenating Commands

- **Command Groups**

  A command group is a group of commands that have common compound headers arranged in a hierarchy. A command group may contain sub-groups.

  Example Group of commands related to acquisition

  ```
  :ACQuire:AVERage:COUNt
  :ACQuire:MODE
  :ACQuire:AVERage:EWEight
  :ACQuire:CLOCK
  :ACQuire:RLENgth
  :ACQuire:COUNt
  ```

- **When Concatenating Commands of the Same Group**

  The DL850/DL850V stores the hierarchical level of the command that is currently being executed and processes the next command on the assumption that it belongs to the same level. Therefore, the common header section can be omitted for commands that belong to the same group.

  Example
  ```
  :ACQuire:MODE NORMal;
  COUNt 1<PMT>
  ```

- **When Concatenating Commands of Different Groups**

  If the subsequent command does not belong to the same group, place a colon in front of the header (cannot be omitted).

  Example
  ```
  :ACQuire:MODE NORMal;:DISPlay:
  FORMat SINGle<PMT>
  ```

- **When Concatenating Simple Headers**

  If a simple header follows another command, place a colon in front of the simple header (cannot be omitted).

  Example
  ```
  :ACQuire:MODE NORMal;:
  STARt<PMT>
  ```

- **When Concatenating Common Commands**

  Common commands that are defined in IEEE 488.2-1992 are independent of hierarchy. There is no need to use a colon.

  Example
  ```
  :ACQuire:MODE NORMal;*CLS;
  COUNt 1<PMT>
  ```

- **When Separating Commands with <PMT>**

  If you separate two commands with a terminator, two program messages will be sent. Therefore, the common header must be specified for each command even if commands belonging to the same command group are being concatenated.

  Example
  ```
  :ACQuire:MODE NORMal<PMT>:
  ACQuire:COUNt 1<PMT>
  ```

Upper-Level Query

An upper-level query is a query that is made by appending a question mark to the highest level command of a group. The controller can receive all of the settings in a group collectively by executing an upper-level query. Some upper-level queries of a group, which may be comprised of more than three hierarchical levels, can cause the DL850/DL850V to transmit all the lower level settings.

Example :
```
CHANnel1?<PMT> -> :CHANNEL1:
DISPLAY ON;LABEL "CH1 ";
COUPLING DC;POSITION 0.00;
PROBE 10;VDIV 50.0E+00;
BWIDTH FULL;OFFSET 0.0E+00;
LSCALE:MODE 0
```

The response to an upper-level query can be sent back to the DL850/DL850V as a program message. This enables the settings that were present when the upper-level query was made to be reproduced later on. However, some upper-level queries do not return setup data that is not currently in use. Exercise caution because not all of a group's information is necessarily returned in a response.

Header Interpretation Rules

The DL850/DL850V interprets the header that it receives according to the rules below.

- Mnemonics are not case sensitive.
  Example `CURSor` can be written as `cursor` or `Cursor`.
- The lower-case characters can be omitted.
  Example `CURSor` can be written as `CURSO` or `CURS`.
- The question mark at the end of a header indicates that it is a query. You cannot omit the question mark.
  Example The shortest abbreviation for `CURSor?` is `CURS?`.
- If the <x> (value) at the end of a mnemonic is omitted, it is interpreted as a 1.
  Example If you write `CHAN` for `CHANnel<x>`, `CHANnel1` is specified.
- Parts of commands and parameters enclosed in square brackets ([ ]) can be omitted.
  Example `TRIGger[:SIMPle]:LEVel` can be written as `TRIG:LEV`.
  However, the last section enclosed in brackets cannot be omitted in an upper-level query.
  Example `TRIGger?` and `TRIGger:SIMPle?` are different queries.

4.3 Response

Response

When the controller sends a query with a question mark, the DL850/DL850V returns a response message to the query. The DL850/DL850V returns response messages in one of the following two forms.

- **Response Consisting of a Header and Data**
  Responses that can be used as program messages without any changes are returned with command headers attached.
  Example `:ACQUire:MODE?<PMT> ->`
  `:ACQUire:MODE NORMAL<RMT>`
- **Response Only Consisting of Data**
  Responses that cannot be used as program messages unless changes are made (query-only commands) are returned without headers. However, there are query-only commands whose responses the DL850/DL850V will attach headers to.
  Example `:MEASure:CHANnel1:PTOPeak:`
  `VALue?<PMT> -> 10.0E+00<RMT>`

If You Want the DL850/DL850V to Return Responses without Headers

You can configure the DL850/DL850V so that even responses that have both headers and data are returned without headers. Use the `COMMunicate:HEADer` command for this purpose.

Abbreviated Form

The DL850/DL850V normally returns response headers with the lower-case section removed. You can configure the DL850/DL850V so that full headers are returned. Use the `COMMunicate:VERBose` command for this purpose. The sections enclosed in braces ([ ]) are also omitted in the abbreviated form.

4.4 Data

Data

Data contains conditions and values that are written after the header. A space separates the data from the header. Data is grouped as follows:

Data	Description
<Decimal>	A value expressed in decimal notation (Example: Probe attenuation for CH1 `-> CHANnel1:PROBe 100`)
<Voltage><Time> <Frequency><Current>	A physical value Time-axis range `-> TIMebase:TDIV 1US`)
<Register>	A register value expressed as binary, octal, decimal or hexadecimal (Example: Extended event register value `-> STATUS:EESE #HFE`)
<Character data>	Predefined character string (mnemonic). Select from the available strings in braces. (Example: Select the input coupling of CH1 `-> CHANnel1:COUPling {AC|DC|DC50|GND}`)
<Boolean>	Indicates on and off. Specify ON, OFF, or a value (Example: Turn on the CH1 display `-> CHANnel1:DISPlay ON`)
<String data>	User-defined string (Example: Comment attached to screen data output `-> HCOPy:COMMent "ABCDEF"`)
<Filename>	Indicates a file name. (Example: Save file name `-> FILE:SAVE:WAVeform:NAME "CASE1"`)
<Block data>	Data that contains 8-bit values (Example: Response to acquired waveform data `-> #800000010ABCDEFGHIJ`)

<Decimal>

<Decimal> indicates a value expressed as a decimal number, as shown in the table below. Decimal values are written in the NR form as specified in ANSI X3.42-1975.

Symbol	Description	Examples		
<NR1>	Integer	`125`	`-1`	`+1000`
<NR2>	Fixed point number	`125.0`	`-.90`	`+001.`
<NR3>	Floating-point number	`125.0E+0`	`-9E-1`	`+.1E4`
<NRf>	Any form from <NR1> to <NR3>			

- The DL850/DL850V can receive decimal values that are sent from the controller in any form, from <NR1> to <NR3>. This is expressed as <NRf>.
- The DL850/DL850V returns a response to the controller in one of the forms from <NR1> to <NR3> depending on the query. The same form is used regardless of the size of the value.
- For the <NR3> form, the plus sign after the “E” can be omitted. You cannot omit the minus sign.
- If a value outside the setting range is entered, the value is adjusted to the closest value within the range.
- If a value has more significant digits than are available, the value will be rounded.

<Voltage>, <Time>, <Frequency>, <Current>

<Voltage>, <Time>, <Frequency>, and <Current> indicate decimal values that have physical significance. A <Multiplier> or <Unit> can be attached to the <NRf> form that was described earlier. The following types of expressions are possible.

Form	Example
<NRf><Multiplier><Unit>	`5MV`
<NRf><Unit>	`5E-3V`
<NRf><Multiplier>	`5M`
<NRf>	`5E-3`

<Multiplier>

<Multipliers> that you can use are indicated in the following table.

Symbol	Word	Multiplier
EX	Exa	10^{18}
PE	Peta	10^{15}
T	Tera	10^{12}
G	Giga	10^{9}
MA	Mega	10^{6}
K	Kilo	10^{3}
M	Milli	10^{-3}
U	Micro	10^{-6}
N	Nano	10^{-9}
P	Pico	10^{-12}
F	Femto	10^{-15}
A	Atto	10^{-18}

<Unit>

<Units> that you can use are indicated in the following table.

Symbol	Word	Description
V	Volt	Voltage
S	Second	Time
HZ	Hertz	Frequency
MHZ	Megahertz	Frequency
A	Ampere	Current

- <Multiplier> and <Unit> are not case sensitive.
- “U” is used to indicate micro (”μ”).
- “MA” is used for Mega to distinguish it from Milli. Megahertz, which is expressed as “MHZ,” is an exception. Therefore, “M (Milli)” cannot be used for frequencies.
- If both <Multiplier> and <Unit> are omitted, the default unit is used.
- Response messages are always expressed in the <NR3> form. Response messages are returned using the default unit without the <Multiplier> or <Unit>.

<Register>

<Register> is an integer that can be expressed in decimal, hexadecimal, octal, or binary notation. It is used when each bit of the value has a particular meaning. The following types of expressions are possible.

Form	Example
<NRf>	1
#H <Hexadecimal value made up of the digits 0 to 9 and A to F>	#H0F
#Q <Octal value made up of the digits 0 to 7>	#Q777
#B <Binary value made up of the digits 0 and 1>	#B001100

- <Register> is not case sensitive.
- Response messages are always expressed in the <NR1> form.

<Character Data>

<Character data> is a predefined character string (mnemonics). It is mainly used to indicate that an option listed as a character string in braces must be selected and entered. The data interpretation rules are the same as those described in “Header Interpretation Rules” on page 4-4.

Form	Example
{AC|DC|GND}	AC

- As with the header, the COMMunicate:VERBose command can be used to select whether to return the response in the full form or in the abbreviated form.
- The COMMunicate:HEADer setting does not affect <character data>.

<Boolean>

<Boolean> is data that indicates on or off. The following types of expressions are possible.

Form	Examples
{ON|OFF|<NRf>}	ON OFF 1 0

- When <Boolean> is expressed in the <NRf> form, OFF is selected if the rounded integer value is 0, and ON is selected for all other cases.
- A response message is always returned with a 1 if the value is ON and with a 0 if the value is OFF.

<String data>

<String data> is not a predefined character string like <character data>. It can be any character string. The character string must be enclosed in single quotation marks (') or double quotation marks (").

Form	Example
<String data>	'ABC' "IEEE488.2-1987"

- If a character string contains a double quotation mark ("), the double quotation mark is expressed as two consecutive quotation marks (""). This rule also applies to single quotation marks.
- A response message is always enclosed in double quotation marks (").
- <String data> is any character string. Therefore, the DL850/DL850V assumes that the remaining program message units are part of the character string if no single (') or double quotation mark (") is encountered. As a result, no error is detected if a quotation mark is omitted.

<Filename>

<Filename> is data that indicates a file name. The following types of expressions are possible.

Form	Example
{<NRf>|<Character data>|<String data>}	1 CASE "CASE"

<NRf> is rounded to an 8-digit integer and converted to ASCII code. The result is the file name (example: 1 becomes "00000001"). Negative values are not allowed.

- The first 12 characters of <character data> or the first 16 characters of <string data> are the file name.
- Response messages are always expressed in the <string data> form.
- For information about the number of characters in a file name expressed in the <string data form>, see the *DL850/DL850V User’s Manual*.

<Block data>

<Block data> is any 8-bit data. It is only used in response messages on the DL850/DL850V. The syntax is as follows:

Form	Example
#N <N-digit decimal number> <data byte sequence>	#800000010ABCDEFGHIJ

- #N
  Indicates that the data is <block data>.“N” indicates the number of succeeding data bytes (digits) in ASCII code.
- <N-digit decimal number>
  Indicates the number of bytes of data (example: 00000010 = 10 bytes).

<Data byte sequence>

Expresses the actual data (example: ABCDEFGHIJ).

- Data is comprised of 8-bit values (0 to 255). This means that the ASCII code “0AH” which stands for “NL” can also be included in the data. Hence, care must be taken when programming the controller.

4.5 Synchronization with the Controller

Overlap Commands and Sequential Commands

There are two types of commands: overlap and sequential. The execution of one overlap command can start before the execution of the previous overlap command is completed.

If you specify V/div and send the next program message to query the result, the DL850/DL850V always returns the most recent setting (5 V in this case).

```
:CHANnel1:VDIV 5V;VDIV?<PMT>
```

This is because the next command is forced to wait until the processing of `CHANnel1:VDIV` is completed. This type of command is called a sequential command.

Let us assume you send the next program message when you want to load a file and query the V/div value of the result.

```
:FILE:LOAD:SETup:EXECute "CASE1";:
CHANnel1:VDIV?
```

In this case, `CHANnel1:VDIV?` is executed before the loading of the file is completed, and the V/div value that is returned is the value before the file is loaded.

Overlapping refers to the act of executing the next command before the processing of the current command is completed, such as in the command `FILE:LOAD:SETup:EXECute "CASE1"`. A command that operates in this way is called an overlap command. You can prevent overlapping by using the following methods.

Synchronizing to Overlap Commands

- **Using a *WAI Command**

  A `*WAI` command holds the subsequent commands until the overlap command is completed.

  Example
  ```
  :COMMunicate:OPSE #H0040;:
  FILE:LOAD:SETup:
  EXECute "CASE1";*WAI;:
  CHANnel1:VDIV?<PMT>
  ```

  The `COMMunicate:OPSE` command is used to select which command to apply `*WAI` to. Here, it is applied to the media access command.

  `*WAI` is executed before `CHANnel1:VDIV?`, so `CHANnel1:VDIV?` is not executed until the file loading is completed.

- **Using the COMMunicate:OVERlap command**

  The `COMMunicate:OVERlap` command enables (or disables) overlapping.

  Example
  ```
  :COMMunicate:OVERlap #HFFBF;:
  FILE:LOAD:SETup:
  EXECute "CASE1";:CHANnel1:
  VDIV?<PMT>
  ```

  `COMMunicate:OVERlap #HFFBF` enables overlapping for commands other than media access. Because overlapping of file loading is disabled, `FILE:LOAD:SETup:EXECute "CASE1"` operates in the same way as a sequential command. Thus, `CHANnel1:VDIV?` is not executed until file loading is completed.

- **Using the *OPC Command**

  The `*OPC` command sets the OPC bit, which is bit 0 in the standard event register (see page 6-4), to 1 when the overlapping is completed.

  Example
  ```
  :COMMunicate:OPSE #H0040;*ESE 1;
  *ESR?;*SRE 32;:FILE:LOAD:SETup:
  EXECute "CASE1";*OPC<PMT>
  (Read the response to *ESR?)
  (Wait for a service request)
  :CHANnel1:VDIV?<PMT>
  ```

  The `COMMunicate:OPSE` command is used to select which command to apply `*OPC` to. Here, it is applied to the media access command.

  `*ESE 1` and `*SRE 32` indicate that a service request is only generated when the OPC bit is 1.

  `*ESR?` clears the standard event register.

  In the example above, `CHANnel1:VDIV?` is not executed until a service request is generated.

- **Using the *OPC? Query**
  The *OPC? query generates a response when an overlapping operation is completed.
  Example :COMMunicate:OPSE #H0040;:
  FILE:LOAD:SETup:
  EXECute "CASE1";*OPC?<PMT>
  (Read the response to *OPC?)
  :CHANnel1:VDIV?<PMT>
  The COMMunicate:OPSE command is used to select which command to apply *OPC? to. Here, it is applied to the media access command.
  Because *OPC? does not generate a response until the overlapping operation is completed, the file loading will have been completed by the time the response to *OPC? is read.

Note

Most commands are sequential commands. Overlap commands are indicated as such in chapter 5. All other commands are sequential commands.

Achieving Synchronization without Using Overlap Commands

Even with sequential commands, synchronization with non-communication events such as triggers is sometimes required to correctly query the measured data.
For example, if the following program message is transmitted to query waveform data acquired with the trigger mode set to single, the WAVeform:SEND? command may be executed regardless of whether the acquisition has been completed and may result in a command execution error.
TRIGger:MODE SINGle;:STARt;:WAVeform:SEND?<PMT>
If this happens, you must use the following method to synchronize to the end of waveform acquisition.

- **Using the STATus:CONDition? query**
  STATus:CONDition? is used to query the contents of the condition register (see page 6-5). You can determine whether waveform acquisition is in progress by reading bit 0 in the condition register. If the bit is 1, waveform acquisition is in progress. If the bit is 0, waveform acquisition is not in progress.
  Example TRIGger:MODE SINGle;:STARt<PMT>
  :STATus:CONDition?<PMT>
  (Read the response. If bit 0 is 1, return to the previous command.)
  :WAVeform:SEND?<PMT>
  WAVeform:SEND? is not executed until bit 0 in the condition register becomes 0.

- **Using the Extended Event Register**
  The changes in the condition register can be reflected in the extended event register (see page 6-5).
  Example :STATus:FILTer1 FALL;:STATus:
  EESE 1;EESR?;*SRE 8;:TRIGger:
  MODE SINGle;:STARt<PMT>
  (Read the response to STATus:EESR?)
  (Wait for a service request)
  :WAVeform:SEND?<PMT>
  The STATus:FILTer1 FALL command sets the transition filter so that bit 0 in the extended event (FILTer1) is set to 1 when bit 0 in the condition register changes from 1 to 0.

  The STATus:EESE 1 command is used to only change the status byte based on bit 0 in the extended event register.
  The STATus:EESR? command is used to clear the extended event register.
  The *SRE 8 command is used to generate service requests based only on the changes in the extended event register bits.
  The WAVeform:SEND? command is not executed until a service request is generated.

- **Using the COMMunicate:WAIT command**
  The COMMunicate:WAIT command is used to wait for a specific event to occur.
  Example :STATus:FILTer1 FALL;:STATus:
  EESR?;:TRIGger:
  MODE SINGle<PMT>
  (Read the response to STATus:EESR?)
  :COMMunicate:WAIT 1;:WAVeform:
  SEND?<PMT>
  For a description of STATus:FILTer1 FALL and STATus:EESR?, see the previous section about the extended event register.
  The COMMunicate:WAIT 1 command specifies that the program will wait for bit 0 in the extended event register to be set to 1.
  WAVeform:SEND? is not executed until bit 0 in the extended event register becomes 1.

5.1 List of Commands

Command	Function	Page
ACQuire Group		
:ACQuire?	Queries all waveform acquisition settings.	5-25
:ACQuire:AVERage?	Queries all averaging settings.	5-25
:ACQuire:AVERage:COUNt	Sets or queries the number of waveform acquisitions to perform during averaging.	5-25
:ACQuire:AVERage:EWEight (Exponent Weight)	Sets or queries the attenuation constant of exponential averaging.	5-25
:ACQuire:CLOCk	Sets or queries the time base (internal or external clock).	5-25
:ACQuire:COUNt	Sets or queries the number of waveform acquisitions to perform in Normal mode.	5-25
:ACQuire:MODE	Sets or queries the waveform acquisition mode.	5-25
:ACQuire:PROTate	Sets or queries the pulse/rotate setting to use during external clock input.	5-25
:ACQuire:RLENgth	Sets or queries the record length.	5-25
:ACQuire:RTOut?	Queries all hard-disk-recording settings.	5-25
:ACQuire:RTOut:ANAMing	Sets or queries the hard-disk-recording auto naming feature.	5-26
:ACQuire:RTOut:COMMent	Sets or queries the hard-disk-recording comment.	5-26
:ACQuire:RTOut:FILename	Sets or queries the hard-disk-recording file name.	5-26
:ACQuire:RTOut:MODE	Sets or queries whether hard disk recording is enabled.	5-26
:ACQuire:RTOut:DIVide:MODE	Sets or queries whether divided recording is enabled for hard disk recording.	5-26
:ACQuire:RTOut:DIVide:NUMBer	Sets or queries the number of divisions when divided recording is performed during hard disk recording.	5-26
ASETup Group		
:ASETup?	Queries all auto setup settings.	5-27
:ASETup:EXECute	Executes auto setup.	5-27
:ASETup:TARGet	Sets or queries the channel to perform auto setup on.	5-27
:ASETup:UNDO	Undoes auto setup.	5-27
CALibrate Group		
:CALibrate?	Queries all calibration settings.	5-28
:CALibrate[:EXECute]	Executes calibration.	5-28
:CALibrate:MODE	Sets or queries the auto calibration mode.	5-28
CAPTure Group		
:CAPTure?	Queries all dual capture settings.	5-29
:CAPTure:ACTion?	Queries all dual-capture-action settings.	5-29
:CAPTure:ACTion:BUZZer	Sets or queries whether a beep is sounded as one of the capture actions.	5-29
:CAPTure:ACTion:MAIL?	Queries all dual-capture-action e-mail transmission settings.	5-29
:CAPTure:ACTion:MAIL:COUNt	Sets or queries the upper limit of e-mail transmissions to perform as a capture action.	5-29
:CAPTure:ACTion:MAIL:MODE	Sets or queries whether e-mail is transmitted as a capture action.	5-29
:CAPTure:ACTion:SAVE:ANAMing	Sets or queries the auto file naming method.	5-29
:CAPTure:ACTion:SAVE?	Queries all dual-capture-action data save settings.	5-29
:CAPTure:ACTion:SAVE[:MODE]	Sets or queries whether captured data is saved to a storage medium as a capture action.	5-29
:CAPTure:ACTion:SAVE:NAME	Sets or queries the file name that is used when captured data is saved to a storage medium.	5-29
:CAPTure:ACTion:SAVE:CDIRectory	Sets the current directory on the storage medium where the screen capture is saved to as a capture action.	5-29
:CAPTure:ACTion:SAVE:TYPE	Sets or queries the data format that is used when captured data is saved to a storage medium.	5-30
:CAPTure:BITDisp	Sets or queries whether captured event waveforms are displayed.	5-30
:CAPTure:CAPNum? MAXimum	Queries the largest number of the captured waveforms.	5-30
:CAPTure:CAPNum? MINimum	Queries the smallest number of the captured waveforms.	5-30
:CAPTure:FORMat	Sets or queries the display format of the capture window.	5-30

Command	Function	Page
:CAPTure:GROup<x1>?	Queries all group display settings of the capture window.	5-30
:CAPTure:GROup<x1>:TRACe<x2>	Sets or queries the displayed waveform in the specified group display of the capture window.	5-30
:CAPTure:MCMode (Main Capture Mode)	Sets or queries the main capture mode during dual capture.	5-30
:CAPTure:MODE	Sets or queries whether the dual capture mode is enabled.	5-30
:CAPTure:RLENgth	Sets or queries the dual capture length.	5-30
:CAPTure:TDIV	Sets or queries the dual capture T/DIV setting.	5-31
:CAPTure:WINDow?	Queries all dual-capture-display-window settings.	5-31
:CAPTure:WINDow:MAG	Sets or queries the dual-capture-window zoom factor as a time.	5-31
:CAPTure:WINDow:MODE	Sets or queries whether the dual capture window is turned on.	5-31
:CAPTure:WINDow:CAPNum	Sets or queries the number of the waveform that is displayed in the dual capture window.	5-31
:CAPTure:WINDow:POSition	Sets or queries the dual-capture-window horizontal position.	5-31

CHANnel Group

:CHANnel<x>?	Queries all vertical axis settings of a channel.	5-32
:CHANnel<x>:ACCL?	Queries all the settings of an installed acceleration/voltage module.	5-32
:CHANnel<x>:ACCL:BIAS	Sets or queries whether the bias current supply to the acceleration sensors of an installed acceleration/voltage module is on.	5-32
:CHANnel<x>:ACCL:BWIDth	Sets or queries the filter when the input coupling of an installed acceleration/voltage module is set to acceleration.	5-32
:CHANnel<x>:ACCL:COUPling	Sets or queries the input coupling setting of an installed acceleration/voltage module.	5-32
:CHANnel<x>:ACCL:GAIN	Sets or queries the gain when the input coupling of an installed acceleration/voltage module is set to acceleration.	5-32
:CHANnel<x>:ACCL:POSition	Sets or queries the vertical position when the input coupling of an installed acceleration/voltage module is set to acceleration.	5-32
:CHANnel<x>:ACCL:SCALe	Sets or queries the upper and lower limits of the screen when the input coupling of an installed acceleration/voltage module is set to acceleration.	5-32
:CHANnel<x>:ACCL:SENSitivity	Sets or queries the sensitivity when the input coupling of an installed acceleration/voltage module is set to acceleration.	5-33
:CHANnel<x>:ACCL:UNIT	Sets or queries the unit string when the input coupling of an installed acceleration/voltage module is set to acceleration.	5-33
:CHANnel<x>:ACCL:VARiable	Sets or queries the scale setup mode when the input coupling of an installed acceleration/voltage module is set to acceleration.	5-33
:CHANnel<x>:ACCL:ZOOM	Sets or queries the vertical zoom factor when the input coupling of an installed acceleration/voltage module is set to acceleration.	5-33
:CHANnel<x>:CAN?	Queries all the settings of a CAN bus monitor module.	5-33
:CHANnel<x>:CAN:OSOut?	Queries all the one-shot output settings of a CAN bus monitor module.	5-33
:CHANnel<x>:CAN:OSOut:EXECute (One Shot Out)	Executes one-shot output on a CAN bus monitor module.	5-33
:CHANnel<x>:CAN:OSOut:DATA<x>	Sets or queries the data frame value used by a CAN bus monitor module when it performs one-shot output.	5-33
:CHANnel<x>:CAN:OSOut:DLC (Data Length Code)	Sets or queries the byte size of the data section of the data frame used for one-shot output.	5-33
:CHANnel<x>:CAN:OSOut:FRAMe	Sets or queries the format (data or remote) of the frame used for one-shot output.	5-33
:CHANnel<x>:CAN:OSOut:MFORmat (Message Format)	Sets or queries the message format (standard or extended) of the frame used for one-shot output.	5-34
:CHANnel<x>:CAN:OSOut:MID (Message ID)	Sets or queries the message ID of the frame used for one-shot output.	5-34
:CHANnel<x>:CAN:PORT?	Queries all the port settings of a CAN bus monitor module.	5-34
:CHANnel<x>:CAN:PORT:BRATe	Sets or queries the bitrate of a CAN bus signal port.	5-34
:CHANnel<x>:CAN:PORT:BSNum (Bit sample num)	Sets or queries the number of sample points for each bit of a CAN bus signal port.	5-34
:CHANnel<x>:CAN:PORT:LONLy	Sets or queries the Listen Only setting for a CAN bus monitor module port.	5-34
:CHANnel<x>:CAN:PORT:SJW (Sync jump width)	Sets or queries the resynchronization jump width (also referred to as the sync jump width).	5-34
:CHANnel<x>:CAN:PORT:SPOint	Sets or queries the sample point for each bit of a CAN bus signal port.	5-34

Command	Function	Page
:CHANnel<x>:CAN:PORT:TERMinator	Sets or queries whether the 124 Ω terminator between CAN_H and CAN_L on the CAN bus line is ON or OFF.	5-34
:CHANnel<x>:CAN:SCHannel<x>?	Queries all the settings of a sub channel on a CAN bus monitor module.	5-34
:CHANnel<x>:CAN:SCHannel<x>:INPut	Sets or queries whether a sub channel on a CAN bus monitor module is ON or OFF.	5-35
:CHANnel<x>:CAN:SCHannel<x>:MFORmat	Sets or queries the message format of a sub channel on a CAN bus monitor module.	5-35
:CHANnel<x>:CAN:SCHannel<x>:MID	Sets or queries the message ID of a sub channel on a CAN bus monitor module.	5-35
:CHANnel<x>:CAN:SCHannel<x>:SBIT (Start BIT)	Sets or queries the bit number of the starting position of a sub channel on a CAN bus monitor module.	5-35
:CHANnel<x>:CAN:SCHannel<x>:BICount (Bit Count)	Sets or queries the bit length of a sub channel on a CAN bus monitor module.	5-35
:CHANnel<x>:CAN:SCHannel<x>:BORDer (Byte Order)	Sets or queries the endianness used to save to internal memory the data of a sub channel on a CAN bus module.	5-35
:CHANnel<x>:CAN:SCHannel<x>:VTYPe (Value Type)	Sets or queries the data type of a sub channel on a CAN bus monitor module.	5-35
:CHANnel<x>:CAN:SCHannel<x>:BCOunt (Byte Count)	Sets or queries the byte count of a sub channel on a CAN bus monitor module.	5-35
:CHANnel<x>:CAN:SCHannel<x>:FACTor	Sets or queries the scaling constant (value per pit) of a sub channel on a CAN bus monitor module.	5-36
:CHANnel<x>:CAN:SCHannel<x>:OFFSet	Sets or queries the scaling constant (offset value) of a sub channel on a CAN bus monitor module.	5-36
:CHANnel<x>:CAN:SCHannel<x>:SCALe	Sets or queries the display range values (upper and lower limits) of a sub channel on a CAN bus monitor module.	5-36
:CHANnel<x>:CAN:SCHannel<x>:POSition	Sets or queries the vertical position of a sub channel on a CAN bus monitor module.	5-36
:CHANnel<x>:CAN:SCHannel<x>:UNIT	Sets or queries the unit of data of a sub channel on a CAN bus monitor module.	5-36
:CHANnel<x>:CAN:SCHannel<x>:BLABel<x>	Sets or queries a bit label of a sub channel on a CAN bus monitor module when the sub channel's data type is logic.	5-36
:CHANnel<x>:CAN:SCHannel<x>:LABel	Sets or queries the label of a sub channel on a CAN bus monitor module.	5-37
:CHANnel<x>:DISPlay	Sets or queries whether the channel is displayed.	5-37
:CHANnel<x>:FREQ?	Queries all the settings of an installed frequency module.	5-37
:CHANnel<x>:FREQ:INPut?	Queries all the input settings of an installed frequency module.	5-37
:CHANnel<x>:FREQ:INPut:BWIDth	Sets or queries the input signal bandwidth limit of an installed frequency module.	5-37
:CHANnel<x>:FREQ:INPut:CELimination	Sets or queries the input signal chattering elimination of an installed frequency module.	5-37
:CHANnel<x>:FREQ:INPut:COUPling	Sets or queries the input coupling setting of an installed frequency module.	5-38
:CHANnel<x>:FREQ:INPut:HYSTeresis	Sets or queries the hysteresis of an installed frequency module.	5-38
:CHANnel<x>:FREQ:INPut:PRESet	Sets or queries the preset setting of an installed frequency module.	5-38
:CHANnel<x>:FREQ:INPut:PROBe	Sets or queries the probe attenuation of an installed frequency module.	5-38
:CHANnel<x>:FREQ:INPut:PULLup	Sets or queries whether pull-up is turned on for an installed frequency module.	5-38
:CHANnel<x>:FREQ:INPut:SLOPe	Sets or queries the input slope of an installed frequency module.	5-38
:CHANnel<x>:FREQ:INPut:THReshold	Sets or queries the threshold level of an installed frequency module.	5-38
:CHANnel<x>:FREQ:INPut:VRANge	Sets or queries the voltage range of an installed frequency module.	5-38
:CHANnel<x>:FREQ:LSCale?	Queries all the linear scaling settings of an installed frequency module.	5-39
:CHANnel<x>:FREQ:LSCale:AVALue	Sets or queries the scaling coefficient A that is used during linear scaling for an installed frequency module.	5-39
:CHANnel<x>:FREQ:LSCale:BVALue	Sets or queries the scaling coefficient B that is used during linear scaling for an installed frequency module.	5-39
:CHANnel<x>:FREQ:LSCale:GETMeasure	Executes the measurement of the linear scaling P1X or P2X value of an installed frequency module.	5-39
:CHANnel<x>:FREQ:LSCale:MODE	Sets or queries the linear scale mode of an installed frequency module.	5-39
:CHANnel<x>:FREQ:LSCale:{P1X|P1Y|P2X|P2Y}	Sets or queries the linear scaling P1X, P1Y, P2X, or P2Y value of an installed frequency module.	5-39

5.1 List of Commands

Command	Function	Page
:CHANnel<x>:FREQ:LSCale:UNIT	Sets or queries the linear-scaling unit string of an installed frequency module.	5-39
:CHANnel<x>:FREQ:OFFSet	Sets or queries the offset of an installed frequency module.	5-40
:CHANnel<x>:FREQ:POSition	Sets or queries the vertical position for an installed frequency module.	5-40
:CHANnel<x>:FREQ:SCALe	Sets or queries the upper and lower limits of the screen for an installed frequency module.	5-40
:CHANnel<x>:FREQ:SETup?	Queries all the FV settings of an installed frequency module.	5-40
:CHANnel<x>:FREQ:SETup:CFRequency	Sets or queries the center frequency of an installed frequency module.	5-40
:CHANnel<x>:FREQ:SETup:DECeleration	Sets or queries whether deceleration prediction is turned on for an installed frequency module.	5-40
:CHANnel<x>:FREQ:SETup:DPULse	Sets or queries the distance per pulse of an installed frequency module.	5-40
:CHANnel<x>:FREQ:SETup:FILTer?	Queries all the filter settings of an installed frequency module.	5-40
:CHANnel<x>:FREQ:SETup:FILTer:SMOothing?	Queries all the smoothing settings of an installed frequency module.	5-41
:CHANnel<x>:FREQ:SETup:FILTer:SMOothing:MODE	Sets or queries whether smoothing is turned on for an installed frequency module.	5-41
:CHANnel<x>:FREQ:SETup:FILTer:SMOothing:VALue	Sets or queries the moving average order of smoothing of an installed frequency module.	5-41
:CHANnel<x>:FREQ:SETup:FILTer:PAVerage?	Queries all the pulse averaging settings of an installed frequency module.	5-41
:CHANnel<x>:FREQ:SETup:FILTer:PAVerage:MODE	Sets or queries whether pulse averaging is turned on for an installed frequency module.	5-41
:CHANnel<x>:FREQ:SETup:FILTer:PAVerage:VALue	Sets or queries the number of pulses to average over for an installed frequency module.	5-41
:CHANnel<x>:FREQ:SETup:FUNCtion	Sets or queries the measurement mode of an installed frequency module.	5-41
:CHANnel<x>:FREQ:SETup:LRESet	Sets or queries whether over-limit reset is turned on for an installed frequency module.	5-42
:CHANnel<x>:FREQ:SETup:MPULse	Sets or queries whether the measurement pulse is positive or negative for an installed frequency module.	5-42
:CHANnel<x>:FREQ:SETup:PROTate	Sets or queries the number of pulses per rotation of an installed frequency module.	5-42
:CHANnel<x>:FREQ:SETup:RESet	Resets the pulse count of an installed frequency module.	5-42
:CHANnel<x>:FREQ:SETup:STOPpredict	Sets or queries whether stop prediction is turned on for an installed frequency module.	5-42
:CHANnel<x>:FREQ:SETup:TUNit	Sets or queries the time unit (when measuring velocity) of an installed frequency module.	5-42
:CHANnel<x>:FREQ:SETup:UNIT	Sets or queries the pulse integration unit of an installed frequency module.	5-42
:CHANnel<x>:FREQ:SETup:UPULse	Sets or queries the unit/pulse setting of an installed frequency module.	5-42
:CHANnel<x>:FREQ:SETup:VUNit	Sets or queries the velocity unit of an installed frequency module.	5-42
:CHANnel<x>:FREQ:VDIV	Sets or queries the Value/Div setting of an installed frequency module.	5-43
:CHANnel<x>:FREQ:ZOOM	Sets or queries the vertical zoom factor of an installed frequency module.	5-43
:CHANnel<x>:LABel	Sets or queries the waveform label of a channel.	5-43
:CHANnel<x>:LOGic?	Queries all settings of an installed logic input module.	5-43
:CHANnel<x>:LOGic:BMAPping	Sets or queries the bit display method of the specified logic channel.	5-43
:CHANnel<x>:LOGic:{BIT1|...|BIT8}?	Queries all settings of each bit of an installed logic input module.	5-43
:CHANnel<x>:LOGic:{BIT1|...|BIT8}:DISPlay	Sets or queries whether the display of each bit is turned on for the specified logic channel.	5-43
:CHANnel<x>:LOGic:{BIT1|...|BIT8}:LABel	Sets or queries the label of each bit for the specified logic channel.	5-43
:CHANnel<x>:LOGic:{BIT1|...|BIT8}:CELimination	Sets or queries the chattering elimination for the specified bit of the specified logic channel.	5-44
:CHANnel<x>:LOGic:POSition	Sets or queries the vertical position of the specified logic channel.	5-44
:CHANnel<x>:LOGic:ZOOM	Sets or queries the vertical zoom factor of an installed module that can perform analog voltage measurements.	5-44
:CHANnel<x>:MODUle?	Queries the module that is installed in the channel.	5-44
:CHANnel<x>:STRain?	Queries all settings of an installed strain module.	5-44
:CHANnel<x>:STRain:BALance?	Queries whether balancing will be performed on an installed strain module.	5-44
:CHANnel<x1>:STRain:BALance:CHANnel<x2>	Sets or queries the channels that balancing will be performed on for an installed strain module.	5-44

Command	Function	Page
:CHANnel<x>:STRain:BALance:EXECute	Executes strain balancing on an installed strain module.	5-44
:CHANnel<x>:STRain:BWIDth	Sets or queries the filter of an installed strain module.	5-45
:CHANnel<x>:STRain:EXCitation	Sets or queries the bridge voltage of an installed strain module.	5-45
:CHANnel<x>:STRain:GFACtor	Sets or queries the gauge factor of an installed strain module.	5-45
:CHANnel<x>:STRain:INVert	Sets or queries whether the display is inverted (ON) or not (OFF) for an installed strain module.	5-45
:CHANnel<x>:STRain:LSCale?	Queries all the linear scaling settings of an installed strain module.	5-45
:CHANnel<x>:STRain:LSCale:AVALue	Sets or queries the scaling coefficient A that is used during linear scaling for an installed strain module.	5-45
:CHANnel<x>:STRain:LSCale:BVALue	Sets or queries the scaling coefficient B that is used during linear scaling for an installed strain module.	5-45
:CHANnel<x>:STRain:LSCale:DISPlaytype?	Queries all the linear-scaling display-mode settings of an installed strain module.	5-45
:CHANnel<x>:STRain:LSCale:DISPlaytype:MODE	Sets or queries the display mode that is used during linear scaling for an installed strain module.	5-45
:CHANnel<x>:STRain:LSCale:DISPlaytype:DECimalnum	Sets or queries the decimal place that is used during linear scaling when the display mode is set to Float for an installed strain module.	5-46
:CHANnel<x>:STRain:LSCale:DISPlaytype:SUBunit	Sets or queries the unit prefix that is used during linear scaling when the display mode is set to Float for an installed strain module.	5-46
:CHANnel<x>:STRain:LSCale:GETMeasure	Executes the measurement of the linear scaling P1X or P2X value of an installed strain module.	5-46
:CHANnel<x>:STRain:LSCale:MODE	Sets or queries the linear scale mode of an installed strain module.	5-46
:CHANnel<x>:STRain:LSCale:{P1X|P1Y|P2X|P2Y}	Sets or queries the linear scaling P1X, P1Y, P2X, or P2Y value of an installed strain module.	5-46
:CHANnel<x>:STRain:LSCale:SHUNt	Executes shunt calibration.	5-46
:CHANnel<x>:STRain:LSCale:UNIT	Sets or queries the linear-scaling unit string of an installed strain module.	5-46
:CHANnel<x>:STRain:RANGe	Sets or queries the measurement range of an installed strain module.	5-46
:CHANnel<x>:STRain:SCALe	Sets or queries the upper and lower limits of the screen for an installed strain module.	5-47
:CHANnel<x>:STRain:UNIT	Sets or queries the unit of an installed strain module.	5-47
:CHANnel<x>:TEMPerature?	Queries all the settings of a module if the module in the specified channel can measure temperature.	5-47
:CHANnel<x>:TEMPerature:BURNout	Sets or queries whether burnout will be detected when temperature measurements are performed.	5-47
:CHANnel<x>:TEMPerature:BWIDth	Sets or queries the bandwidth limit that is used when temperature measurements are performed.	5-47
:CHANnel<x>:TEMPerature:COUPling	Sets or queries the input coupling setting of an installed module that can measure temperature.	5-47
:CHANnel<x>:TEMPerature:RJC	Sets or queries the RJC that is used when temperature measurements are performed.	5-47
:CHANnel<x>:TEMPerature:SCALe	Sets or queries the upper and lower limits of the screen that are used when temperature measurements are performed.	5-47
:CHANnel<x>:TEMPerature:TYPE	Sets or queries the thermocouple type that is used when temperature measurements are performed.	5-47
:CHANnel<x>:TEMPerature:UNIT	Sets or queries the unit that is used when temperature measurements are performed.	5-48
:CHANnel<x>:VOLTage?	Queries all settings of an installed module that can perform analog voltage measurements.	5-48
:CHANnel<x>[:VOLTage]:BWIDth	Sets or queries the bandwidth limit of an installed module that can perform analog voltage measurements.	5-48
:CHANnel<x>[:VOLTage]:COUPling	Sets or queries the input coupling of an installed module that can perform analog voltage measurements.	5-48
:CHANnel<x>[:VOLTage]:INVert	Sets or queries whether the display is inverted (ON) or not (OFF) for an installed module that can perform analog voltage measurements.	5-48
:CHANnel<x>[:VOLTage]:LSCale?	Queries all the linear scaling settings of an installed module that can perform analog voltage measurements.	5-48
:CHANnel<x>[:VOLTage]:LSCale:AVALue	Sets or queries the scaling coefficient A that is used during linear scaling for an installed module that can perform analog voltage measurements.	5-48
:CHANnel<x>[:VOLTage]:LSCale:BVALue	Sets or queries the offset value B that is used during linear scaling for an installed module that can perform analog voltage measurements.	5-49
:CHANnel<x>[:VOLTage]:LSCale:DISPlaytype?	Queries all the linear-scaling display-mode settings of an installed module that can perform analog voltage measurements.	5-49

Command	Function	Page
:CHANnel<x>[:VOLTage]:LSCale:DISPlaytype:MODE	Sets or queries the display mode that is used during linear scaling for an installed module that can perform analog voltage measurements.	5-49
:CHANnel<x>[:VOLTage]:LSCale:DISPlaytype:DECimalnum	Sets or queries the decimal place that is used during linear scaling when the display mode is set to Float for an installed module that can perform analog voltage measurements.	5-49
:CHANnel<x>[:VOLTage]:LSCale:DISPlaytype:SUBunit	Sets or queries the unit prefix that is used during linear scaling when the display mode is set to Float for an installed module that can perform analog voltage measurements.	5-49
:CHANnel<x>[:VOLTage]:LSCale:GETMeasure	Executes the measurement of the linear scaling P1X or P2X value of an installed module that can perform analog voltage measurements.	5-49
:CHANnel<x>[:VOLTage]:LSCale:MODE	Sets or queries the linear scale mode of an installed module that can perform analog voltage measurements.	5-49
:CHANnel<x>[:VOLTage]:LSCale:{P1X|P1Y|P2X|P2Y}	Sets or queries the linear scaling P1X, P1Y, P2X, or P2Y value of an installed module that can perform analog voltage measurements.	5-50
:CHANnel<x>[:VOLTage]:LSCale:UNIT	Sets or queries the linear-scaling unit string of an installed module that can perform analog voltage measurements.	5-50
:CHANnel<x>[:VOLTage]:OFFSet	Sets or queries the offset voltage of an installed module that can perform analog voltage measurements.	5-50
:CHANnel<x>[:VOLTage]:POSition	Sets or queries the vertical position of an installed module that can perform analog voltage measurements.	5-50
:CHANnel<x>[:VOLTage]:PROBe	Sets or queries the probe type of an installed module that can perform analog voltage measurements.	5-50
:CHANnel<x>[:VOLTage]:SCALe	Sets or queries the upper and lower limits of the screen for an installed module that can perform analog voltage measurements.	5-50
:CHANnel<x>[:VOLTage]:VARiable	Sets or queries how the vertical scale is set for an installed module that can perform analog voltage measurements.	5-50
:CHANnel<x>[:VOLTage]:VDIV	Sets or queries the V/div setting of an installed module that can perform analog voltage measurements.	5-51
:CHANnel<x>[:VOLTage]:ZOOM	Sets or queries the vertical zoom factor of an installed module that can perform analog voltage measurements.	5-51
:CHANnel<x>[:VOLTage]:SCHannel<x>?	Queries all settings of an installed 16-CH Voltage Input Module.	5-51
:CHANnel<x>[:VOLTage]:SCHannel<x>:LABel	Sets or queries the label string of the specified sub channel for the specified 16-CH Voltage Input Module.	5-51
:CHANnel<x>[:VOLTage]:SCHannel<x>:BWIDth	Sets or queries the bandwidth limit of an installed 16-CH Voltage Input Module.	5-51
:CHANnel<x>[:VOLTage]:SCHannel<x>:COUPling	Sets or queries the input coupling setting of an installed 16-CH Voltage Input Module.	5-51
:CHANnel<x>[:VOLTage]:SCHannel<x>:INVert	Sets or queries whether the display is inverted (ON) or not (OFF) for an installed 16-CH Voltage Input Module.	5-52
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:LSCale?	Queries all the linear scaling settings of an installed 16-CH Voltage Input Module.	5-52
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:LSCale:AVALue	Sets or queries the scaling coefficient A that is used during linear scaling for an installed 16-CH Voltage Input Module.	5-52
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:LSCale:BVALue	Sets or queries the offset value B that is used during linear scaling for an installed 16-CH Voltage Input Module.	5-52
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:LSCale:DISPlaytype?	Queries all the linear-scaling display-mode settings of an installed 16-CH Voltage Input Module.	5-52
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:LSCale:DISPlaytype:MODE	Sets or queries the display mode that is used during linear scaling for an installed 16-CH Voltage Input Module.	5-52
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:LSCale:DISPlaytype:DECimalnum	Sets or queries the decimal place that is used during linear scaling when the display mode is set to Float for an installed 16-CH Voltage Input Module.	5-53
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:LSCale:DISPlaytype:SUBunit	Sets or queries the unit prefix that is used during linear scaling when the display mode is set to Float for an installed 16-CH Voltage Input Module.	5-53
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:LSCale:GETMeasure	Executes the measurement of the linear scaling P1X or P2X value of an installed 16-CH Voltage Input Module.	5-53
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:LSCale:MODE	Sets or queries the linear scale mode of an installed 16-CH Voltage Input Module.	5-53

Command	Function	Page
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:LSCale:{P1X|P1Y|P2X|P2Y}	Sets or queries the linear scaling P1X, P1Y, P2X, or P2Y value of an installed 16-CH Voltage Input Module.	5-53
:CHANnel<x1>[:VOLTage]:SCHannel<x2>: LSCale:UNIT	Sets or queries the linear-scaling unit string of an installed 16-CH Voltage Input Module.	5-53
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:OFFSet	Sets or queries the offset voltage of an installed 16-CH Voltage Input Module.	5-54
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:POSition	Sets or queries the vertical position for an installed 16-CH Voltage Input Module.	5-54
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:SCALe	Sets or queries the upper and lower limits of the screen for an installed 16-CH Voltage Input Module.	5-54
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:VARiable	Sets or queries how the vertical scale is set for an installed 16-CH Voltage Input Module.	5-54
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:VDIV	Sets or queries the V/DIV setting of an installed 16-CH Voltage Input Module.	5-54
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:ZOOM	Sets or queries the vertical zoom factor of an installed 16-CH Voltage Input Module.	5-54

CLEar Group

:CLEar	Clears traces.	5-55

COMMunicate Group

:COMMunicate?	Queries all communication settings.	5-56
:COMMunicate:HEADer	Sets or queries whether headers are attached to query responses.	5-56
:COMMunicate:LOCKout	Sets or clears local lockout.	5-56
:COMMunicate:OPSE	Sets or queries the overlap command that is used by the *OPC, *OPC?, and *WAI commands.	5-56
:COMMunicate:OPSR?	Queries the operation pending status register.	5-56
:COMMunicate:OVERlap	Sets or queries the commands that operate as overlap commands.	5-56
:COMMunicate:REMote	Sets or queries whether the DL850/DL850V is in remote or local mode. ON is remote mode.	5-56
:COMMunicate:VERBose	Sets or queries whether query responses are returned in full or abbreviated form.	5-56
:COMMunicate:WAIT	Waits for a specified extended event to occur.	5-56
:COMMunicate:WAIT?	Creates the response that is returned when a specified extended event occurs.	5-57

CURSor Group

:CURSor?	Queries all cursor measurement settings.	5-58
:CURSor:FFT?	Queries all FFT cursor settings.	5-58
:CURSor:FFT:MARKer:M<x>?	Queries all settings of an FFT cursor (marker cursor).	5-58
:CURSor:FFT:MARKer:M<x>:DF<y>?	Queries all ΔF display settings. ΔF is between the FFT cursors (marker cursors).	5-58
:CURSor:FFT:MARKer:M<x>:DF<y>:STATe	Sets or queries whether the ΔF value between the FFT cursors (marker cursors) is displayed.	5-58
:CURSor:FFT:MARKer:M<x>:DF<y>:VALue?	Queries the ΔF value between the FFT cursors (marker cursors).	5-58
:CURSor:FFT:MARKer:M<x>:DY<y>?	Queries all ΔY display settings. ΔY is between the FFT cursors (marker cursors).	5-58
:CURSor:FFT:MARKer:M<x>:DY<y>:STATe	Sets or queries whether the ΔY value between the FFT cursors (marker cursors) is displayed.	5-58
:CURSor:FFT:MARKer:M<x>:DY<y>:VALue?	Queries the ΔY value between the FFT cursors (marker cursors).	5-58
:CURSor:FFT:MARKer:M<x>:POSition	Sets or queries the position of an FFT cursor (marker cursor).	5-58
:CURSor:FFT:MARKer:M<x>:TRACe	Sets or queries the source waveform that you want to measure using the FFT cursor (marker cursor).	5-59
:CURSor:FFT:MARKer:M<x>:F?	Queries all frequency-axis settings of an FFT cursor (marker cursor).	5-59
:CURSor:FFT:MARKer:M<x>:F:STATe	Sets or queries whether the frequency-axis value of an FFT cursor (marker cursor) is displayed.	5-59

Command	Function	Page
:CURSor:FFT:MARKer:M<x>:F:VALue?	Queries the frequency-axis value of an FFT cursor (marker cursor).	5-59
:CURSor:FFT:MARKer:M<x>:Y?	Queries all Y-axis settings of an FFT cursor (marker cursor).	5-59
:CURSor:FFT:MARKer:M<x>:Y:STATe	Sets or queries whether the Y-axis value of an FFT cursor (marker cursor) is displayed.	5-59
:CURSor:FFT:MARKer:M<x>:Y:VALue?	Queries the Y-axis value of an FFT cursor (marker cursor).	5-59
:CURSor:FFT:MARKer:FORM	Sets or queries the form of an FFT cursor (marker cursor).	5-59
:CURSor:FFT:PEAK<x>?	Queries all settings of an FFT cursor (peak cursor).	5-59
:CURSor:FFT:PEAK<x>:F?	Queries all frequency-axis settings of an FFT cursor (peak cursor).	5-59
:CURSor:FFT:PEAK<x>:F:STATe	Sets or queries whether the frequency-axis value of an FFT cursor (peak cursor) is displayed.	5-59
:CURSor:FFT:PEAK<x>:F:VALue?	Queries the frequency-axis value of an FFT cursor (peak cursor).	5-59
:CURSor:FFT:PEAK<x>:Y?	Queries all Y-axis settings of an FFT cursor (peak cursor).	5-60
:CURSor:FFT:PEAK<x>:Y:STATe	Sets or queries whether the Y-axis value of an FFT cursor (peak cursor) is displayed.	5-60
:CURSor:FFT:PEAK<x>:Y:VALue?	Queries the Y-axis value of an FFT cursor (peak cursor).	5-60
:CURSor:FFT:TYPE	Sets or queries the FFT cursor type.	5-60
:CURSor[:TY]?	Queries all cursor settings for the T-Y display.	5-60
:CURSor[:TY]:DEGRee?	Queries all angle cursor settings for the T-Y display.	5-60
:CURSor[:TY]:DEGRee:D<X>?	Queries all angle difference (D<x>) settings for the angle cursors on the T-Y display.	5-60
:CURSor[:TY]:DEGRee:D<X>:STATe	Sets or queries whether the angle difference (D<x>) between the angle cursors on the T-Y display is displayed.	5-60
:CURSor[:TY]:DEGRee:D<X>:VALue?	Queries the angle difference (D<x>) between the angle cursors on the T-Y display.	5-60
:CURSor[:TY]:DEGRee:DD?	Queries all angle difference (Δd value) settings. The angle difference is between the angle cursors on the T-Y display.	5-60
:CURSor[:TY]:DEGRee:DD:STATe	Sets or queries whether the angle difference (Δd) value between the angle cursors on the T-Y display is displayed.	5-60
:CURSor[:TY]:DEGRee:DD:VALue?	Queries the angle difference (Δd value) between the angle cursors on the T-Y display.	5-60
:CURSor[:TY]:DEGRee:DY?	Queries all angle difference (ΔY value) settings. The angle difference is between the angle cursors on the T-Y display.	5-60
:CURSor[:TY]:DEGRee:DY:STATe	Sets or queries whether the angle difference (ΔY) value between the angle cursors on the T-Y display is displayed.	5-61
:CURSor[:TY]:DEGRee:DY:VALue?	Queries the angle difference (ΔY value) between the angle cursors on the T-Y display.	5-61
:CURSor[:TY]:DEGRee:POSition<x>	Sets or queries an angle cursor position on the T-Y display.	5-61
:CURSor[:TY]:DEGRee:REFerence<x>	Sets or queries the angle reference start (Ref1) or end (Ref2) point on the T-Y display.	5-61
:CURSor[:TY]:DEGRee:RVALue	Sets or queries an angle cursor's reference angle on the T-Y display.	5-61
:CURSor[:TY]:DEGRee:TRACe	Sets or queries the source waveform that you want to measure using the angle cursors on the T-Y display.	5-61
:CURSor[:TY]:DEGRee:Y<x>?	Queries all measured-value settings for an angle cursor on the T-Y display.	5-61
:CURSor[:TY]:DEGRee:Y<x>:STATe	Sets or queries whether the measured value of an angle cursor on the T-Y display is displayed.	5-61
:CURSor[:TY]:DEGRee:Y<x>:VALue?	Queries the measured value of an angle cursor on the T-Y display.	5-61
:CURSor[:TY]:HORizontal?	Queries all horizontal cursor settings for the T-Y display.	5-61
:CURSor[:TY]:HORizontal:DY?	Queries all ΔY-axis settings of the horizontal cursors on the T-Y display.	5-62
:CURSor[:TY]:HORizontal:DY:STATe	Sets or queries whether the ΔY-axis value between the horizontal cursors on the T-Y display is displayed.	5-62
:CURSor[:TY]:HORizontal:DY:VALue?	Queries the ΔY-axis value of the horizontal cursors on the T-Y display.	5-62
:CURSor[:TY]:HORizontal:POSition<x>	Sets or queries a horizontal cursor position on the T-Y display.	5-62
:CURSor[:TY]:HORizontal:TRACe	Sets or queries the source waveform that you want to measure using the horizontal cursors on the T-Y display.	5-62
:CURSor[:TY]:HORizontal:Y<x>?	Queries all Y-axis settings of the horizontal cursor on the T-Y display.	5-62
:CURSor[:TY]:HORizontal:Y<x>:STATe	Sets or queries whether the Y-axis value for a horizontal cursor on the T-Y display is displayed.	5-62

Command	Function	Page
:CURSor[:TY]:HORizontal:Y<x>:VALue?	Queries the Y-axis value of a horizontal cursor on the T-Y display.	5-62
:CURSor[:TY]:MARKer?	Queries all marker cursor settings for the T-Y display.	5-62
:CURSor[:TY]:MARKer:M<x>?	Queries all settings related to the marker cursor measurement items for the T-Y display.	5-62
:CURSor[:TY]:MARKer:M<x>:DX<y>?	Queries all ΔX value settings. ΔX is between the marker cursors on the T-Y display.	5-62
:CURSor[:TY]:MARKer:M<x>:DX<y>:STATe	Sets or queries whether the ΔX value between the marker cursors on the T-Y display is displayed.	5-62
:CURSor[:TY]:MARKer:M<x>:DX<y>:VALue?	Queries the ΔX value between the marker cursors on the T-Y display.	5-63
:CURSor[:TY]:MARKer:M<x>:DY<y>?	Queries all ΔY value settings. ΔY is between the marker cursors on the T-Y display.	5-63
:CURSor[:TY]:MARKer:M<x>:DY<y>:STATe	Sets or queries whether the ΔY value between the marker cursors on the T-Y display is displayed.	5-63
:CURSor[:TY]:MARKer:M<x>:DY<y>:VALue?	Queries the ΔY value between the marker cursors on the T-Y display.	5-63
:CURSor[:TY]:MARKer:M<x>:POSition	Sets or queries a marker cursor position on the T-Y display.	5-63
:CURSor[:TY]:MARKer:M<x>:TRACe	Sets or queries the source waveform that you want to measure using the marker cursors on the T-Y display.	5-63
:CURSor[:TY]:MARKer:M<x>:X?	Queries all X-axis settings for a marker cursor on the T-Y display.	5-63
:CURSor[:TY]:MARKer:M<x>:X:STATe	Sets or queries whether the X-axis value for a marker cursor on the T-Y display is displayed.	5-63
:CURSor[:TY]:MARKer:M<x>:X:VALue?	Queries the X-axis value of a marker cursor on the T-Y display.	5-63
:CURSor[:TY]:MARKer:M<x>:Y?	Queries all Y-axis settings for a marker cursor on the T-Y display.	5-64
:CURSor[:TY]:MARKer:M<x>:Y:STATe	Sets or queries whether the Y-axis value for a marker cursor on the T-Y display is displayed.	5-64
:CURSor[:TY]:MARKer:M<x>:Y:VALue?	Queries the Y-axis value of a marker cursor on the T-Y display.	5-64
:CURSor[:TY]:MARKer:FORM	Sets or queries the form of the marker cursors on the T-Y display.	5-64
:CURSor[:TY]:TYPE	Sets or queries the cursor type on the T-Y display.	5-64
:CURSor[:TY]:VERTical?	Queries all V cursor settings for the T-Y display.	5-64
:CURSor[:TY]:VERTical:DX?	Queries all ΔX value settings. ΔX is between the V cursors on the T-Y display.	5-64
:CURSor[:TY]:VERTical:DX:STATe	Sets or queries whether the ΔX value between the V cursors on the T-Y display is displayed.	5-64
:CURSor[:TY]:VERTical:DX:VALue?	Queries the ΔX value between the V cursors on the T-Y display	5-64
:CURSor[:TY]:VERTical:DY?	Queries all Y-axis-value settings for V cursors on the T-Y display.	5-64
:CURSor[:TY]:VERTical:DY:STATe	Sets or queries whether the Y-axis value for V cursors on the T-Y display is displayed.	5-64
:CURSor[:TY]:VERTical:DY:VALue?	Queries the Y-axis value of V cursors on the T-Y display.	5-64
:CURSor[:TY]:VERTical:PERDt?	Queries all 1/ΔT value settings. 1/ΔT is between the V cursors on the T-Y display.	5-64
:CURSor[:TY]:VERTical:PERDt:STATe	Sets or queries whether the 1/ΔT value between the V cursors on the T-Y display is displayed.	5-65
:CURSor[:TY]:VERTical:PERDt:VALue?	Queries the 1/ΔT value between the V cursors on the T-Y display	5-65
:CURSor[:TY]:VERTical:POSition<x>	Sets or queries a V cursor position on the T-Y display.	5-65
:CURSor[:TY]:VERTical:TRACe	Sets or queries the source waveform that you want to measure using the V cursors on the T-Y display.	5-65
:CURSor[:TY]:VERTical:X<x>?	Queries all X-axis-value settings for a V cursor on the T-Y display.	5-65
:CURSor[:TY]:VERTical:X<x>:STATe	Sets or queries whether the X-axis value for a V cursor on the T-Y display is displayed.	5-65

Command	Function	Page
:CURSor[:TY]:VERTical:X<x>:VALue?	Queries the X-axis value of a V cursor on the T-Y display.	5-65
:CURSor[:TY]:VERTical:Y<x>?	Queries all Y-axis-value settings for a V cursor on the T-Y display.	5-65
:CURSor[:TY]:VERTical:Y<x>:STATe	Sets or queries whether the Y-axis value for a V cursor on the T-Y display is displayed.	5-65
:CURSor[:TY]:VERTical:Y<x>:VALue?	Queries the Y-axis value (measured value) of a V cursor on the T-Y display.	5-66
:CURSor:XY?	Queries all cursor settings for the X-Y display.	5-66
:CURSor:XY:HORizontal?	Queries all H cursor settings for the X-Y display.	5-66
:CURSor:XY:HORizontal:DY?	Queries all ΔY-axis-value settings. The ΔY-axis value is between the H cursors on the X-Y display.	5-66
:CURSor:XY:HORizontal:DY:STATe	Sets or queries whether the ΔY-axis value between the H cursors on the X-Y display is displayed.	5-66
:CURSor:XY:HORizontal:DY:VALue?	Queries the ΔY-axis value between the H cursors on the X-Y display	5-66
:CURSor:XY:HORizontal:POSition<x>	Sets or queries H cursor positions on the X-Y display.	5-66
:CURSor:XY:HORizontal:TRACe	Sets or queries the source waveform that you want to measure using the H cursors on the X-Y display.	5-66
:CURSor:XY:HORizontal:Y<x>?	Queries all Y-axis-value settings for an H cursor on the X-Y display.	5-66
:CURSor:XY:HORizontal:Y<x>:STATe	Sets or queries whether the Y-axis value for an H cursor on the X-Y display is displayed.	5-66
:CURSor:XY:HORizontal:Y<x>:VALue?	Queries the Y-axis value of an H cursor on the X-Y display	5-66
:CURSor:XY:MARKer:M<x>?	Queries all marker cursor settings for the X-Y display.	5-66
:CURSor:XY:MARKer:M<x>:POSition	Sets or queries a marker cursor's time-axis-equivalent position on the X-Y display.	5-67
:CURSor:XY:MARKer:M<x>:T?	Queries all time value settings for a marker cursor on the X-Y display.	5-67
:CURSor:XY:MARKer:M<x>:T:STATe	Sets or queries whether the time value for a marker cursor on the X-Y display is displayed.	5-67
:CURSor:XY:MARKer:M<x>:T:VALue?	Queries the time value of a marker cursor on the X-Y display.	5-67
:CURSor:XY:MARKer:M<x>:TRACe	Sets or queries the source waveform that you want to measure using the marker cursors on the X-Y display.	5-67
:CURSor:XY:MARKer:M<x>:X?	Queries all X-axis-value settings for a marker cursor on the X-Y display.	5-67
:CURSor:XY:MARKer:M<x>:X:STATe	Sets or queries whether the X-axis value for a marker cursor on the X-Y display is displayed.	5-67
:CURSor:XY:MARKer:M<x>:X:VALue?	Queries the X-axis value of a marker cursor on the X-Y display.	5-67
:CURSor:XY:MARKer:M<x>:Y?	Queries all Y-axis-value settings for a marker cursor on the X-Y display.	5-67
:CURSor:XY:MARKer:M<x>:Y:STATe	Sets or queries whether the Y-axis value for a marker cursor on the X-Y display is displayed.	5-67
:CURSor:XY:MARKer:M<x>:Y:VALue?	Queries the Y-axis value of a marker cursor on the X-Y display.	5-67
:CURSor:XY:MARKer:FORM	Sets or queries the form of the marker cursors on the X-Y display.	5-68
:CURSor:XY:TYPE	Sets or queries the cursor type on the X-Y display.	5-68
:CURSor:XY:VERTical?	Queries all V cursor settings for the X-Y display.	5-68
:CURSor:XY:VERTical:DX?	Queries all ΔX-axis settings. ΔX is between the V cursors on the X-Y display.	5-68
:CURSor:XY:VERTical:DX:STATe	Sets or queries whether the ΔX-axis value between the V cursors on the X-Y display is displayed.	5-68
:CURSor:XY:VERTical:DX:VALue?	Queries the ΔX-axis value between the V cursors on the X-Y display	5-68
:CURSor:XY:VERTical:POSition<x>	Sets or queries a V cursor position on the X-Y display.	5-68
:CURSor:XY:VERTical:TRACe	Sets or queries the source waveform that you want to measure using the V cursors on the X-Y display.	5-68
:CURSor:XY:VERTical:X<x>?	Queries all X-axis-value settings for a V cursor on the X-Y display.	5-68
:CURSor:XY:VERTical:X<x>:STATe	Sets or queries whether the X-axis value for a V cursor on the X-Y display is displayed.	5-68
:CURSor:XY:VERTical:X<x>:VALue?	Queries the X-axis value of a V cursor on the X-Y display.	5-68
:CURSor:WAIT?	While the DL850/DL850V is waiting for the specified :CURSor command to finish, it will wait the time specified by the :CURSor:WAIT? timeout value until it begins processing the subsequent command.	5-69

DISPlay Group

Command	Function	Page
:DISPlay?	Queries all display settings.	5-70
:DISPlay:ACCumulate?	Queries all accumulated waveform display settings.	5-70
:DISPlay:ACCumulate:MODE	Sets or queries the accumulated waveform mode.	5-70
:DISPlay:ACCumulate:PERSistence	Sets or queries the number of times that waveforms are accumulated.	5-70
:DISPlay:CINFormation (Channel Information)	Sets or queries the contents of the channel information display.	5-70
:DISPlay:CHANnel<x>:SCHannel<x>:COLor	Sets or queries a waveform color.	5-70
:DISPlay:COLor:BASecolor	Sets or queries the base color of the screen.	5-70
:DISPlay:DECimation	Sets or queries the number of dots (after decimation) that are used on the dot display.	5-70
:DISPlay:ESIZe (Extra window Size)	Sets or queries the size of the extra window.	5-71
:DISPlay:FORMat	Sets or queries the display format (the number of divisions in the vertical direction).	5-71
:DISPlay:GRATicule	Sets or queries the grid type.	5-71
:DISPlay:GROup<x1>?	Queries all display group settings.	5-71
:DISPlay:GROup<x1>:TRACe<x2>?	Queries all source waveform settings for a display group.	5-71
:DISPlay:GROup<x1>:TRACe<x2>:SOURce	Sets or queries the source that is assigned to the specified source waveform of the specified display group.	5-71
:DISPlay:GROup<x1>:TRACe<x2>:ZNUMber (Zone Number)	Sets or queries the zone number that displays the specified source waveform of the specified display group.	5-71
:DISPlay:INTENsity?	Queries all intensity settings.	5-71
:DISPlay:INTENsity:{CURSor|GRID|MARKer}	Sets or queries the intensity of a display item.	5-71
:DISPlay:INTerpolate	Sets or queries the waveform interpolation method.	5-71
:DISPlay:SDGRoup (Select Display Group)	Sets or queries the number of the display group that is displayed.	5-71
:DISPlay:MAPPing	Sets or queries the waveform mapping mode for when the display format is set to an option other than 1.	5-72
:DISPlay:SVALue (Scale VALUE)	Sets or queries whether scale values are displayed.	5-72
:DISPlay:TLABel (Trace LABEL)	Sets or queries whether waveform labels are displayed.	5-72
:DISPlay:LINDicator (Level Indicator)	Sets or queries whether the right indicator on the TY waveform display is displayed.	5-72
:DISPlay:MODE:EKEY (Esc Key)	Switches the screen display mode.	5-72

EVENt Group

:EVENt?	Queries all event waveform settings.	5-73
:EVENt:DCAPture[:MODE]	Sets or queries whether the event waveform's dual capture events are displayed.	5-73
:EVENt:MEVent:MODE (Manual Event)	Sets or queries whether the event waveform's manual events are displayed.	5-73
:EVENt:MEVent:EXECute	Manually inserts an event.	5-73
:EVENt:MEVent:SLOPe	Sets or queries the manual event (external event terminal) polarity.	5-73

FFT Group

:FFT?	Queries all FFT settings.	5-74
:FFT:AVERage?	Queries all FFT averaging settings.	5-74
:FFT:AVERage:MODE	Sets or queries the FFT averaging mode.	5-74
:FFT:AVERage:TYPE	Sets or queries the averaging domain.	5-74
:FFT:AVERage:COUNt	Sets or queries the average count of linear averaging.	5-74
:FFT:AVERage:EWEight	Sets or queries the attenuation constant of exponential averaging.	5-74
:FFT:POINt	Sets or queries the number of analysis source points of the FFT windows.	5-74
:FFT:STARt	Sets or queries the analysis-source start point of the FFT windows.	5-74
:FFT:WINDow?	Queries all FFT analysis settings.	5-74
:FFT:WINDow:TYPE	Sets or queries the computation window type for FFT analysis.	5-74
:FFT:WINDow:EXPonential?	Queries all FFT exponential window settings.	5-74
:FFT:WINDow:EXPonential:DRATe	Sets or queries the FFT exponential window's damping rate.	5-74
:FFT:WINDow:EXPonential:FORCe<x>	Sets or queries the FFT exponential window's Force1 or Force2 value.	5-75
:FFT:WAVeform<x>?	Queries all FFT window settings.	5-75
:FFT:WAVeform<x>:TYPE	Sets or queries an FFT window's analysis type.	5-75

Command	Function	Page
:FFT:WAVeform<x>:STYPe (Sub Type)	Sets or queries an FFT window's analysis sub type.	5-75
:FFT:WAVeform<x>:MODE	Sets or queries whether an FFT window is displayed.	5-75
:FFT:WAVeform<x>:HAXis	Sets or queries an FFT window's horizontal-axis display method.	5-75
:FFT:WAVeform<x>:VSCale?	Queries all FFT vertical scale settings.	5-75
:FFT:WAVeform<x>:VSCale:MODE (Vertical Scale Mode)	Sets or queries an FFT window's vertical scale mode.	5-75
:FFT:WAVeform<x>:VSCale:CENTer	Sets or queries the center value of an FFT window's vertical scale.	5-75
:FFT:WAVeform<x>:VSCale:SENSitivity	Sets or queries the sensitivity of an FFT window's vertical scale.	5-75
:FFT:WAVeform<x>:SOURce<x>	Sets or queries an FFT window's FFT source channel.	5-76

FILE Group

:FILE?	Queries all storage media settings.	5-77
:FILE:COPY:ABORt	Aborts file copying.	5-77
:FILE:COPY:CDIRectory	Changes the file copy destination directory.	5-77
:FILE:COPY[:EXECute]	Executes file copying. This is an overlap command.	5-77
:FILE:COPY:PATH?	Queries the file copy destination directory.	5-77
:FILE:DELete	Deletes files. This is an overlap command.	5-77
:FILE[:DIRectory]:CDIRectory (Change Directory)	Changes the current directory on the storage medium.	5-77
:FILE[:DIRectory]:DRIVe	Sets the storage medium to perform file operations on.	5-77
:FILE[:DIRectory]:FREE?	Queries the free space on the medium that is being operated on in bytes.	5-77
:FILE[:DIRectory]:MDIRectory (Make Directory)	Creates a directory in the current directory.	5-77
:FILE[:DIRectory]:PATH?	Queries the current directory.	5-78
:FILE:LOAD:{BINary|SETup|SNAP}:ABORt	Aborts the loading of various types of data.	5-78
:FILE:LOAD:{BINary|SETup|SNAP}[:EXECute]	Executes the loading of various types of data. This is an overlap command.	5-78
:FILE:SAVE?	Queries all file name settings for saving files.	5-78
:FILE:SAVE:ANAMing	Sets or queries the auto naming mode for saving files.	5-78
:FILE:SAVE:{ASCii|BINary|FLOat}?	Queries all the settings related to the saving of a specific type of file.	5-78
:FILE:SAVE:{ASCii|BINary|FLOat|MEASure|SETup|SNAP}:ABORt	Aborts the saving of a specific type of file.	5-78
:FILE:SAVE:{ASCii|BINary|FLOat|MEASure|SETup|SNAP}[:EXECute]	Executes the saving of a specific type of file. This is an overlap command.	5-78
:FILE:SAVE:{ASCii|BINary|FLOat}:HISTory	Sets or queries what waveforms the history memory feature will save for a specific type of data.	5-78
:FILE:SAVE:{ASCii|BINary|FLOat}:RANGe	Sets or queries the save range for a specific type of data.	5-78
:FILE:SAVE:{ASCii|BINary|FLOat}:CRANge<x>	Sets or queries the cursor position for when data is to be saved using a cursor range.	5-78
:FILE:SAVE:{ASCii|BINary|FLOat}:STRace:{CHANnel<x>|MATH<x>}	Sets or queries the waveform that will be saved for a specific type of data.	5-78
:FILE:SAVE:{ASCii|BINary|FLOat}:TALL (Trace All)	Sets or queries the selection method for the waveforms that will be saved for a specific type of data.	5-79
:FILE:SAVE:ASCii:INTerval	Sets or queries the data removal interval that is used when saving data in ASCII format.	5-79
:FILE:SAVE:ASCii:TINFormation	Sets or queries whether time information is included when saving data in ASCII format.	5-79
:FILE:SAVE:ASCii:DPOint	Sets or queries the type of decimal point that is used when saving data in ASCII format.	5-79
:FILE:SAVE:ASCii:SCHannel	Sets or queries the writing method of sub channel data when saving data in ASCII format.	5-79
:FILE:SAVE:ASCii:EXTension	Sets or queries the file name extension that is used when saving data in ASCII format.	5-79
:FILE:SAVE:BINary:COMPression	Sets or queries whether waveform data is compressed during saving.	5-79
:FILE:SAVE:COMMent	Sets or queries the comment that will be saved.	5-79
:FILE:SAVE:NAME	Sets or queries the name of the file that will be saved.	5-79

Command	Function	Page
GONogo Group		
:GONogo?	Queries all GO/NO-GO determination settings.	5-80
:GONogo:ACONdition	Sets or queries the GO/NO-GO determination-action condition.	5-80
:GONogo:ACTion?	Queries all settings for the action that is performed when the condition is met and the settings for the condition itself.	5-80
:GONogo:ACTion:BUZZer	Sets or queries whether a beep is sounded when the condition is met.	5-80
:GONogo:ACTion:HCOPy (HardCOPY)	Sets or queries whether a screen capture is printed from the built-in printer when the condition is met.	5-80
:GONogo:ACTion:IMAGe?	Queries all settings for the screen capture that is saved when the condition is met.	5-80
:GONogo:ACTion:IMAGe[:MODE]	Sets or queries whether a screen capture is saved when the condition is met.	5-80
:GONogo:ACTion:IMAGe:CDIRectory	Sets the current directory where the screen capture is saved to when the condition is met.	5-80
:GONogo:ACTion:IMAGe:DRIVe	Sets or queries the medium that the screen capture is saved to when the condition is met.	5-81
:GONogo:ACTion:IMAGe:PATH?	Queries the path on the storage medium that a screen capture is saved to when the condition is met.	5-81
:GONogo:ACTion:MAIL?	Queries all settings for sending e-mail when the condition is met.	5-81
:GONogo:ACTion:MAIL:COUNt	Sets or queries the number of times that e-mail is sent when the condition is met.	5-81
:GONogo:ACTion:MAIL:MODE	Sets or queries whether e-mail is sent when the condition is met.	5-81
:GONogo:ACTion:SAVE[:MODE]	Sets or queries whether waveforms are saved to the storage medium when the condition is met.	5-81
:GONogo:ACTion:SAVE:CDIRectory	Sets the current directory on the storage medium where waveform data is saved to when the condition is met.	5-81
:GONogo:ACTion:SAVE:DRIVe	Sets the storage medium that waveform data is saved to when the condition is met.	5-82
:GONogo:ACTion:SAVE:PATH?	Queries the path on the storage medium that waveform data is saved to when the condition is met.	5-82
:GONogo:ACTion:SAVE:TYPE	Sets or queries the file format that waveforms are saved as on the storage medium when the condition is met.	5-82
:GONogo:COUNt?	Queries the number of GO/NO-GO determinations that were performed.	5-82
:GONogo:LOGic	Sets or queries the GO/NO-GO determination logic.	5-82
:GONogo:MODE	Sets or queries the GO/NO-GO determination mode.	5-82
:GONogo:NGCount?	Queries the GO/NO-GO determination NO-GO count.	5-82
:GONogo:PARameter?	Queries all parameter determination settings.	5-82
:GONogo:PARameter:ITEM<x>?	Queries all settings for the specified waveform parameter for parameter determination.	5-82
:GONogo:PARameter:ITEM<x>:CAUSe?	Queries whether the specified waveform parameter for parameter determination is the cause of a NO-GO judgment.	5-82
:GONogo:PARameter:ITEM<x>:MODE	Sets or queries the specified waveform parameter's reference condition for parameter determination.	5-82
:GONogo:PARameter:ITEM<x>:TRACe	Sets or queries the specified waveform parameter's source waveform for parameter determination.	5-83
:GONogo:PARameter:ITEM<x>:TYPE?	Queries, for parameter determination, the specified waveform parameter's measurement item and upper and lower limits.	5-83
:GONogo:PARameter:ITEM<x>:TYPE:<Parameter>	Sets or queries the upper and lower limits of the measurement item for the specified waveform parameter.	5-83
:GONogo:PARameter:ITEM<x>:VALue?	Queries the measured value of the specified waveform parameter.	5-83
:GONogo:REMote	Sets or queries the remote mode of GO/NO-GO determination.	5-83
:GONogo:SEQuence	Sets or queries the action mode of GO/NO-GO determination.	5-83
:GONogo:TRANge	Sets or queries the determination range of GO/NO-GO determination.	5-83
:GONogo:ZONE?	Queries all waveform zone determination settings.	5-83
:GONogo:ZONE:PATTern<x>?	Queries all settings for the specified determination pattern for waveform zone determination.	5-83
:GONogo:ZONE:PATTern<x>:CAUSe?	Queries whether the specified determination pattern for waveform zone determination is the cause of a NO-GO judgment.	5-83
:GONogo:ZONE:PATTern<x>:MODE	Sets or queries the specified determination pattern's reference condition for waveform zone determination.	5-83
:GONogo:ZONE:PATTern<x>:TRACe	Sets or queries the specified determination pattern's determination waveform for waveform zone determination.	5-84

Command	Function	Page
:GONogo:ZONE:PATTern<x>:ZONE	Sets or queries the specified determination pattern's source waveform zone data for waveform zone determination.	5-84
:GONogo:WAIT?	Waits for the completion of GO/NO-GO determination by using a timeout value.	5-84

HCOPy Group

:HCOPy?	Queries all screen capture data output settings.	5-85
:HCOPy:COMMent	Sets or queries the screen comment.	5-85
:HCOPy:DIRection	Sets or queries the data output destination.	5-85
:HCOPy:EXECute	Executes data output.	5-85
:HCOPy:NETPrint?	Queries all network printer output settings.	5-85
:HCOPy:NETPrint:TONE	Sets or queries whether data will be printed in color from the network printer.	5-85
:HCOPy:NETPrint:TYPE	Sets or queries the network printer output command type.	5-85

HISTory Group

:HISTory?	Queries all of the settings for the history feature.	5-86
:HISTory:ABORt	Aborts the history search.	5-86
:HISTory:CLEar	Clears the data of all history waveforms.	5-86
:HISTory:DISPlay	Sets or queries the history start and end numbers that will be displayed.	5-86
:HISTory:DMODe (Display MODE)	Sets or queries the history waveform display mode.	5-86
:HISTory:EXECute	Executes the history waveform search. This is an overlap command.	5-86
:HISTory:PARameter?	Queries all history-waveform parameter-search settings.	5-86
:HISTory:PARameter:ITEM<x>?	Queries all the specified parameter's settings for history-waveform parameter searches.	5-86
:HISTory:PARameter:ITEM<x>:CONDition	Sets or queries the specified parameter's reference condition for history-waveform parameter searches.	5-86
:HISTory:PARameter:ITEM<x>:SOURce	Sets or queries the specified parameter's source trace for history-waveform parameter searches.	5-86
:HISTory:PARameter:ITEM<x>:TYPE?	Queries, for history-waveform parameter searches, the specified parameter's automatically measured item and upper and lower limits.	5-86
:HISTory:PARameter:ITEM<x>:TYPE:<Parameter>	Sets or queries the specified parameter's upper and lower limits for history-waveform parameter searches.	5-87
:HISTory:PARameter:ITEM<x>:VALue?	Queries the specified parameter's measured value for history-waveform parameter searches.	5-87
:HISTory:PARameter:LOGic	Sets or queries the logic to apply to history waveform searches.	5-87
:HISTory:PARameter:TRANge	Sets or queries the determination range of history waveform parameter searches.	5-87
:HISTory:RECord	Sets or queries the source record.	5-87
:HISTory:RECord? MINimum	Queries the minimum record number.	5-87
:HISTory:SMODe	Sets or queries the history waveform search mode.	5-87
:HISTory:DATE?	Queries the trigger date of the data at the specified record number.	5-87
:HISTory:TIME?	Queries the trigger time of the data at the specified record number.	5-88
:HISTory:ZONE?	Queries all history-waveform zone-search settings.	5-88
:HISTory:ZONE:EDIT<x>?	Queries all settings for the specified search zone.	5-88
:HISTory:ZONE:EDIT<x>:CONDition	Sets or queries the specified search zone's search condition.	5-88
:HISTory:ZONE:EDIT<x>:SOURce	Sets or queries the specified search zone's source waveform.	5-88
:HISTory:ZONE:LOGic	Sets or queries the logic condition of history-waveform zone searches.	5-88

IMAGe Group

:IMAGe?	Queries all screen capture data output settings.	5-89
:IMAGe:BACKground	Sets or queries the screen capture background (png).	5-89
:IMAGe:COMMent	Sets or queries the screen comment.	5-89
:IMAGe:EXECute	Saves the screen capture data.	5-89
:IMAGe:FORMat	Sets or queries the screen capture output format.	5-89
:IMAGe:SAVE?	Queries all file output settings.	5-89
:IMAGe:SAVE:ANAMing	Sets or queries the setting of the auto naming feature for saving files.	5-89
:IMAGe:SAVE:CDIRectory	Changes the output destination directory.	5-89
:IMAGe:SAVE:DRIVe	Sets the output destination medium.	5-89
:IMAGe:SAVE:FRAMe	Sets or queries whether a white frame is attached to the saved screen capture's image.	5-89

Command	Function	Page
:IMAGe:SAVE:NAME	Sets or queries the name of the file that will be saved.	5-89
:IMAGe:SAVE:PATH?	Queries the current directory.	5-89
:IMAGe:SEND?	Queries the screen capture data.	5-89
:IMAGe:TONE	Sets or queries the color tone of the screen capture data that will be saved.	5-89

INITialize Group

:INITialize:EXECute	Initializes the settings. This is an overlap command.	5-90
:INITialize:UNDO	Undoes the setting initialization.	5-90

LSTart Group

:LSTart (Log STart)	Starts waveform acquisition immediately.	5-91
:LSTart?	Starts waveform acquisition immediately, and waits for acquisition to complete.	5-91

MATH Group

:MATH<x>?	Queries all computation settings.	5-92
:MATH<x>:AVERage?	Queries all averaging computation settings.	5-92
:MATH<x>:AVERage:CCOunt	Sets or queries the cycle count of cycle averaging.	5-92
:MATH<x>:AVERage:COUNt	Sets or queries the average count of linear averaging.	5-92
:MATH<x>:AVERage:EWEight	Sets or queries the attenuation constant of exponential averaging.	5-92
:MATH<x>:AVERage:MODE	Sets or queries the averaging mode.	5-92
:MATH<x>:AVERage:TYPE	Sets or queries the averaging domain.	5-92
:MATH<x>:BINary?	Queries all binary computation settings.	5-92
:MATH<x>:BINary:CHANnel<x>:SCHannel<x>[:THReshold]	Sets or queries the threshold level of the specified channel for binary computations.	5-93
:MATH<x>:CONStant<x>	Sets or queries a constant for user-defined computation.	5-93
:MATH<x>:DEFine	Sets or queries an expression for user-defined computation.	5-93
:MATH<x>:DISPlay	Sets or queries whether computations will be performed.	5-93
:MATH<x>:ESHift:CHANnel<x>:SCHannel<x>[:COUNt]	Sets or queries the amount of phase shift in the Shift computation when an external clock is used.	5-93
:MATH<x>:EXPonential?	Queries all exponential window settings.	5-93
:MATH<x>:EXPonential:DRATe	Sets or queries the exponential window's damping rate.	5-94
:MATH<x>:EXPonential:FORCe<x>	Sets or queries the exponential window's FORCe1 or FORCe2 value.	5-94
:MATH<x>:FFT?	Queries all FFT computation settings.	5-94
:MATH<x>:FFT:POINt	Sets or queries the number of FFT points.	5-94
:MATH<x>:FFT:WINDow	Sets or queries the time window for FFT computations.	5-94
:MATH<x>:FILTer<x>?	Queries all digital filter settings.	5-94
:MATH<x>:FILTer<x>:BAND	Sets or queries a digital filter band.	5-94
:MATH<x>:FILTer<x>:CUToff<x>	Sets or queries a cutoff frequency of a digital filter.	5-94
:MATH<x>:FILTer<x>:TYPE	Sets or queries the type of a digital filter.	5-94
:MATH<x>:LABel	Sets or queries a computed waveform label.	5-95
:MATH<x>:MODE	Sets or queries whether the DL850/DL850V is in computation mode.	5-95
:MATH<x>:MREFerence	Sets or queries the computation range.	5-95
:MATH<x>:OPERation	Sets or queries the computation type.	5-95
:MATH<x>:SOURce<x>	Sets or queries a sub channel's computation type.	5-95
:MATH<x>:SCALe?	Queries all scaling settings.	5-95
:MATH<x>:SCALe:MODE	Sets or queries a scale mode.	5-95
:MATH<x>:SCALe:VALue	Sets or queries a set of upper and lower limits for manual scaling.	5-95
:MATH<x>:SHIFt:CHANnel<x>:SCHannel<x>[:TIME]	Sets or queries the amount of phase shift in the Shift computation when the internal clock is used.	5-95
:MATH<x>:UNIT	Sets or queries a unit string that is attached to computation results.	5-96

Command	Function	Page
MEASure Group		
:MEASure?	Queries all the settings for automated measurement of waveform parameters.	5-97
:MEASure:CHANnel<x>:SCHannel<x>?	Sets or queries whether all the waveform parameters of the specified channel are ON or OFF.	5-97
:MEASure:CHANnel<x>:SCHannel<x>:ALL	Sets all the measurement items of the specified channel to ON or OFF.	5-97
:MEASure:CHANnel<x>:SCHannel<x>:CDEStination (CopyDestination)	Sets or queries whether the copy destination channel is ON or OFF for when parameter measurement items are copied between channels.	5-97
:MEASure:CHANnel<x>:SCHannel<x>:COPY	Copies all the measurement item ON/OFF settings from one specified channel to another specified channel.	5-97
:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DELay?	Queries all delay settings.	5-97
:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DELay:COUNt?	Queries the statistics count of the delay between channels.	5-98
:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DELay:{MAXimum|MEAN|MINimum|SDEViation}?	Queries a statistic of the delay between channels.	5-98
:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DELay:MEASure?	Queries all the settings for a source waveform for measuring the delay between channels of a waveform.	5-98
:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DELay:MEASure:COUNt	Sets or queries the number of source waveforms at which delay between channels will be measured for a waveform.	5-98
:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DELay:MEASure:SLOPe	Sets or queries a source waveform slope that will be used to measure delay between channels for a waveform.	5-98
:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DELay:REFerence?	Queries all reference waveform settings used to measure the delay between channels for the specified waveform.	5-98
:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DELay:REFerence:COUNt	Sets or queries the edge detection count of the reference waveform used to measure the delay between channels for the specified waveform.	5-98
:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DELay:REFerence:SLOPe	Sets or queries the edge detection slope of the reference waveform used to measure the delay between channels for the specified waveform.	5-99
:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DELay:REFerence:SOURce	Sets or queries whether to set the reference point for measuring the delay between channels for a waveform to a trigger point or to a waveform.	5-99
:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DELay:REFerence:TRACe	Sets or queries the reference waveform trace used to measure the delay between channels for a waveform.	5-99
:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DELay:STATe	Sets or queries the display format of the delay parameters of the specified channel.	5-99
:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DELay:VALue?	Queries a measured delay value of the specified waveform's parameter.	5-99
:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DPRoximal?	Queries all distal, mesial, and proximal settings.	5-99
:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DPRoximal:MODE	Sets or queries the distal, mesial, and proximal point mode setting.	5-99
:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DPRoximal:PERCent	Sets or queries the distal, mesial, and proximal points as percentages.	5-100
:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DPRoximal:UNIT	Sets or queries the distal, mesial, and proximal points as units.	5-100

Command	Function	Page
:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:METHod	Sets or queries the modes of a set of high and low points (rising-time and falling-time measurement references).	5-100
:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:<Parameter>?	Queries the setting of a waveform parameter (measurement item).	5-100
:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:<Parameter>:COUNt?	Queries the count of measured values for cyclic statistical processing of a waveform parameter.	5-100
:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:<Parameter>:{MAXimum|MEAN|MINimum|SDEViation}?	Queries a cyclic statistical processing value of a waveform parameter.	5-101
:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:<Parameter>STATe	Sets or queries whether the specified waveform's waveform parameter (measurement item) is ON or OFF.	5-101
:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:<Parameter>VALue?	Queries the measured value of a waveform parameter.	5-101
:MEASure:{XY<x>}?	Queries all the settings for automated measurement of XY waveform parameters.	5-101
:MEASure:{XY<x>}:<Parameter>?	Queries all the settings of a waveform parameter (measurement item).	5-101
:MEASure:{XY<x>}:<Parameter>:COUNt?	Queries the count of measured values for cyclic statistical processing of a waveform parameter.	5-102
:MEASure:{XY<x>}:<Parameter>:{MAXimum|MEAN|MINimum|SDEViation}?	Queries a cyclic statistical processing value of a waveform parameter.	5-102
:MEASure:{XY<x>}:<Parameter>:STATe	Sets or queries whether the specified waveform's waveform parameter (measurement item) is ON or OFF.	5-102
:MEASure:{XY<x>}:<Parameter>:VALue?	Queries the measured value of a waveform parameter.	5-102
:MEASure:CYCLe?	Queries all cycle statistics settings.	5-102
:MEASure:CYCLe:ABORt	Aborts cyclic statistical processing.	5-102
:MEASure:CYCLe:EXECute	Executes cyclic statistical processing.	5-102
:MEASure:CYCLe:TRACe	Sets or queries the cycle trace of cycle statistics.	5-103
:MEASure:HISTory:ABORt	Aborts the statistical processing of history waveforms.	5-103
:MEASure:HISTory:EXECute	Executes the statistical processing of history waveforms.	5-103
:MEASure:MODE	Sets or queries the measure mode.	5-103
:MEASure:ONECycle	Sets or queries whether one cycle mode is ON or OFF.	5-103
:MEASure:TRANge (Time Range)	Sets or queries the waveform parameter measurement range.	5-103
:MEASure:WAIT?	Waits for the completion of measurement by using a timeout valuer.	5-103

MONitor Group

:MONitor:ASENd?	Sends the numeric monitor data (in ASCII format) of all channels.	5-104
:MONitor:ASENd:CHANnel<x>[:SCHannel<x>]?	Sends the numeric monitor data (in ASCII format) of the specified channel.	5-104
:MONitor:BITS:CHANnel<x>[:SCHannel<x>]?	Queries the effective bit length of the specified channel's data.	5-104
:MONitor:BYTeorder	Sets or queries the transmission byte order for data formats that are 2 bytes or longer.	5-104
:MONitor:FORMat:CHANnel<x>[:SCHannel<x>]	Sets or queries the transmission data format.	5-104
:MONitor:OFFSet:CHANnel<x>[:SCHannel<x>]?	Queries the offset that is used when the specified channel's numeric monitor data is converted to a physical value.	5-104
:MONitor:GAIN:CHANnel<x>[:SCHannel<x>]?	Queries the gain that is used when the specified channel's numeric monitor data is converted to a physical value.	5-104
:MONitor:RANGe:CHANnel<x>[:SCHannel<x>]?	Queries the range that is used when the specified channel's numeric monitor data is converted to a physical value.	5-105
:MONitor:SEND:ALL?	Sends the numeric monitor data (in binary format).	5-105
:MONitor:SEND:CHANnel<x>[:SCHannel<x>]?	Sends the numeric monitor data (in binary format) of the specified channel.	5-105
:MONitor:VERBose	Set whether to include labels and units in the response to the MONitor:ASENd? command.	5-105
:MONitor:LATCh:ASENd?	Sends the numeric monitor data (in ASCII format) of all channels.	5-105
:MONitor:LATCh:ASENd:CHANnel<x>[:SCHannel<x>]?	Sends the numeric monitor data (in ASCII format) of the specified channel.	5-105
:MONitor:LATCh:EXECute	Latches the monitor data.	5-105
:MONitor:LATCh:SEND:{ALL|CHANnel<x>[:SCHannel<x>]}?	Sends numeric monitor data.	5-105

Command	Function	Page
MTRigger Group		
:MTRigger	Manually triggers the DL850/DL850V.	5-106
RECall Group		
:RECall:SETup<x>:EXECute	Recalls setup data from an internal memory area.	5-107
SEARch Group		
:SEARch?	Queries all search settings.	5-108
:SEARch:ABORt	Aborts the search.	5-108
:SEARch:EDGE?	Queries all edge search settings.	5-108
:SEARch:EDGE:COUNt	Sets or queries the edge search count.	5-108
:SEARch:EDGE:HYSTeresis	Sets or queries the edge search determination-level hysteresis.	5-108
:SEARch:EDGE:LEVEl	Sets or queries the edge search determination level.	5-108
:SEARch:EDGE:[LOGic:]BIT<x>	Sets or queries the edge polarity of the specified bit for when the edge search is performed on logic traces.	5-108
:SEARch:EDGE:SLOPe	Sets or queries the edge polarity for when the edge search is performed on traces other than logic traces.	5-108
:SEARch:EDGE:SOURce	Sets or queries the trace to perform the edge search on.	5-108
:SEARch:EVENt?	Queries all event search settings.	5-108
:SEARch:EVENt:COUNt	Sets or queries the event search event number.	5-109
:SEARch:EVENt:SOURce	Sets or queries the event search source event.	5-109
:SEARch:EXECute	Executes the search. This is an overlap command.	5-109
:SEARch:LSTate?	Queries all logic pattern search settings.	5-109
:SEARch:LSTate:BIT<x>	Sets or queries the level of the specified bit for logic pattern searches.	5-109
:SEARch:LSTate:SOURce	Sets or queries the trace to perform the logic pattern search on.	5-109
:SEARch:LSTate:COUNt	Sets or queries the logic pattern search count.	5-109
:SEARch:SELect	Sets the search point that is displayed on the zoom window, and queries the zoom position of that search point.	5-109
:SEARch:SELect? MAXimum	Queries the maximum save number, which is the number that is attached to the last position that the search retrieved.	5-109
:SEARch:EPOint (End POint)	Sets or queries the search end position.	5-109
:SEARch:SPOint (Start POint)	Sets or queries the search start position.	5-109
:SEARch:TIME?	Queries all time search settings.	5-110
:SEARch:TIME:TIME	Sets or queries the time of the time search.	5-110
:SEARch:TWINdow (Target WINdow)	Sets or queries the window that search results will be displayed in.	5-110
:SEARch:TYPE	Sets or queries the search type.	5-110
:SEARch:MAG<x>	Sets or queries a zoom waveform magnification.	5-110
:SEARch:POSition<x>	Sets or queries the position of a zoom box.	5-110
:SEARch:TDIV<x>	Sets or queries a zoom waveform T/DIV value.	5-110
SNAP Group		
:SNAP	Takes a snapshot.	5-111
SSTart Group		
:SSTart	Executes the single start operation.	5-112
:SSTart? {<NRf>}	Executes the single start operation and waits for its completion by using a timeout valuer.	5-112
STARt Group		
:STARt	Starts waveform acquisition.	5-113

Command	Function	Page
STATus Group		
:STATus?	Queries all the settings for the communication status feature.	5-114
:STATus:CONDition?	Queries the contents of the condition register.	5-114
:STATus:EESE	Sets or queries the extended event enable register.	5-114
:STATus:EESR?	Queries the contents of the extended event register and clears the register.	5-114
:STATus:ERRor?	Queries the error code and message of the last error that has occurred.	5-114
:STATus:FILTer<x>	Sets or queries the transition filter.	5-114
:STATus:QENable	Sets or queries whether messages other than errors will be stored to the error queue (ON) or not (OFF).	5-114
:STATus:QMESsage	Sets or queries whether message information will be attached to the response to the STAT:ERR? query (ON) or not (OFF).	5-114
STOP Group		
:STOP	Stops waveform acquisition.	5-115
STORe Group		
:STORe?	Queries all the information related to setup data in the internal memory.	5-116
:STORe:SETup<x>?	Queries information about the setup data in the specified location of the internal memory.	5-116
:STORe:SETup<x>:COMMent	Sets or queries the comment for the setup data that is stored to the specified location in the internal memory.	5-116
:STORe:SETup<x>:DATE?	Queries the date and time of the setup data that is stored to the specified location in the internal memory.	5-116
:STORe:SETup<x>:EXECute	Saves setup data to the specified location in the internal memory.	5-116
SYSTem Group		
:SYSTem?	Queries all system settings.	5-117
:SYSTem:CLICk	Sets or queries whether click sounds are produced.	5-117
:SYSTem:CLOCk?	Queries all date/time settings.	5-117
:SYSTem:CLOCk:DATE	Sets or queries the date.	5-117
:SYSTem:CLOCk:FORMat	Sets or queries the date format.	5-117
:SYSTem:CLOCk:MODE	Sets or queries whether the date and time are displayed.	5-117
:SYSTem:CLOCk:SNTP?	Queries all SNTP settings.	5-117
:SYSTem:CLOCk:SNTP:EXECute	Uses SNTP to set the date and time.	5-117
:SYSTem:CLOCk:SNTP:GMTTime	Sets or queries the time difference from GMT when SNTP is being used.	5-117
:SYSTem:CLOCk:TIME	Sets or queries the time.	5-117
:SYSTem:CLOCk:TYPE	Sets or queries whether the date and time are set manually or by using SNTP.	5-117
:SYSTem:IRIG?	Queries all IRIG settings.	5-118
:SYSTem:IRIG:FORMat	Sets or queries the IRIG format for synchronizing the time through IRIG signals.	5-118
:SYSTem:IRIG:MODulation	Sets or queries the IRIG modulation type for synchronizing the time through IRIG signals.	5-118
:SYSTem:IRIG:IMPedance	Sets or queries the input impedance for synchronizing the time through IRIG signals.	5-118
:SYSTem:KEYResponse	Sets or queries the response time of the START/STOP key.	5-118
:SYSTem:KEYProtect?	Queries all key lock settings.	5-118
:SYSTem:KEYProtect:MODE	Sets or queries whether the keys are locked.	5-118
:SYSTem:KEYProtect:TYPE	Sets or queries which keys will be locked.	5-118
:SYSTem:KEYProtect:RTYPe (Release TYPe)	Sets or queries how to release the key lock.	5-118
:SYSTem:KEYProtect:EPASsword (Enter password)	Enters the password to release the key lock.	5-118
:SYSTem:KEYProtect:SPASsword (Set password)	Sets the password that is used to release the key lock.	5-118
:SYSTem:LANGuage	Sets or queries the message language.	5-119
:SYSTem:LCD?	Queries all LCD settings.	5-119
:SYSTem:LCD:AUTO?	Queries all the settings for the feature that automatically turns off the backlight.	5-119
:SYSTem:LCD:AUTO:MODE	Sets or queries whether the feature that automatically turns off the backlight is on.	5-119

Command	Function	Page
:SYSTem:LCD:AUTO:TIME	Sets or queries the amount of time until the backlight is turned off.	5-119
:SYSTem:LCD:BRIGhtness	Sets or queries the LCD brightness.	5-119
:SYSTem:LCD:MODE	Sets or queries whether the backlight is on.	5-119
:SYSTem:LOGic?	Queries all logic display settings.	5-119
:SYSTem:LOGic:NFORmat (Numerical Format)	Sets or queries the numeric display format of logic module data.	5-119
:SYSTem:LOGic:CORDer (Cursor Order)	Sets or queries the bit order that is used when you use cursors to display logic module data as bits.	5-119
:SYSTem:LOGic:BORDer (Bit Order)	Sets or queries the bit order that is used when you display logic module data as waveforms.	5-119
:SYSTem:MLANguage	Sets or queries the menu language.	5-120
:SYSTem:OVERview	Displays the system information.	5-120
:SYSTem:PACTion	Sets or queries whether the action-on-trigger mode will be enabled when the DL850/DL850V turns on (ON) or not (OFF).	5-120
:SYSTem:PSTart	Sets or queries whether waveform acquisition will be started when the DL850/DL850V turns on (ON) or not (OFF).	5-120
:SYSTem:RCMode (Remote Control Mode)	Sets or queries whether remote signals to stop measurements will be ignored (ON) or not (OFF).	5-120
:SYSTem:SCALefont	Sets or queries the font size that is used for waveform labels and scale values.	5-120
:SYSTem:SOITem (Scale On ITEM)	Sets or queries which scales are displayed.	5-120
:SYSTem:STORage:MEDia	Sets or queries the medium that you want to format.	5-120
:SYSTem:STORage:FORMat:EXECute	Formats the specified medium. This is an overlap command.	5-120
:SYSTem:TSYNchro (Time Synchronization)	Sets or queries the time synchronization.	5-120
:SYSTem:USBKeyboard	Sets or queries the USB keyboard type.	5-120

TIMebase Group

:TIMebase?	Queries all time base settings.	5-121
:TIMebase:CHANnel<x>:SRATe?	Queries the sample rate of the specified channel.	5-121
:TIMebase:SOURce	Sets or queries the time base.	5-121
:TIMebase:SRATe	Sets or queries the sample rate.	5-121
:TIMebase:TDIV	Sets or queries the T/DIV value.	5-121

TRIGger Group

:TRIGger?	Queries all trigger settings.	5-122
:TRIGger:ABN? (A -> B(n))	Queries all A->B(n) trigger settings.	5-122
:TRIGger:ABN:COUNt	Sets or queries the number of times condition B must be met for A->B(n) triggers.	5-122
:TRIGger:ACTion?	Queries all action-on-trigger settings.	5-122
:TRIGger:ACTion:BUZZer	Sets or queries whether a beep is sounded as an action.	5-122
:TRIGger:ACTion:HCOPy	Sets or queries whether an screen capture is printed from the built-in printer as an action.	5-122
:TRIGger:ACTion:IMAGe?	Queries all settings for saving screen captures as an action-on-trigger.	5-122
:TRIGger:ACTion:IMAGe[:MODE]	Sets or queries whether a screen capture is saved to the storage medium as an action.	5-122
:TRIGger:ACTion:IMAGe:CDIRectory	Sets or queries the current directory on the storage medium where screen captures are saved to as an action.	5-122
:TRIGger:ACTion:IMAGe:DRIVe	Sets the storage medium that screen captures are saved to as an action.	5-122
:TRIGger:ACTion:IMAGe:PATH?	Queries the path on the storage medium where screen captures are saved to as an action.	5-122
:TRIGger:ACTion:MAIL?	Queries all settings for sending e-mail as an action-on-trigger.	5-122
:TRIGger:ACTion:MAIL:COUNt	Sets or queries the e-mail transmission limit for when e-mail is sent as an action.	5-123
:TRIGger:ACTion:MAIL:MODE	Sets or queries whether e-mail is sent as an action.	5-123
:TRIGger:ACTion:MODE	Sets or queries whether action-on-trigger is used.	5-123
:TRIGger:ACTion:SAVE?	Queries all the settings related to saving data as an action-on-trigger.	5-123
:TRIGger:ACTion:SAVE[:MODE]	Sets or queries whether waveform data is saved to the storage medium as an action.	5-123
:TRIGger:ACTion:SAVE:TYPE	Sets or queries the type of data to save when waveform data is saved to the storage medium as an action.	5-123
:TRIGger:ACTion:SAVE:CDIRectory	Changes the current directory on the storage medium that waveform data is saved to as an action.	5-123

Command	Function	Page
:TRIGger:ACTion:SAVE:DRIVe	Sets the storage medium that waveform data is saved to as an action.	5-123
:TRIGger:ACTion:SAVE:PATH?	Queries the path on the storage medium that waveform data is saved to as an action.	5-123
:TRIGger:ADB? (A Delay B)	Queries all A Delay B trigger settings.	5-123
:TRIGger:ADB:DELay	Sets or queries the delay time for condition B for A Delay B triggers.	5-123
:TRIGger:AND?	Queries all AND trigger settings.	5-123
:TRIGger:AND:CHANnel<x>:BIT<x>	Sets or queries the state of the specified bit of the specified logic channel for AND triggers.	5-124
:TRIGger:AND:CHANnel<x>:SCHannel<x>:BIT<x>	Sets or queries the state of the specified bit of the specified sub channel of the specified logic channel for AND triggers.	5-124
:TRIGger:AND:CHANnel<x>[:CONDition]	Sets or queries the state of the specified channel of the specified logic channel for AND triggers.	5-124
:TRIGger:AND:CHANnel<x>:SCHannel<x>[:CONDition]	Sets or queries the state of the specified sub channel of the specified logic channel for AND triggers.	5-124
:TRIGger:ATRigger?	Queries all condition A settings.	5-124
:TRIGger:ATRigger:CHANnel<x>:BIT<x>	Sets or queries the state of the specified bit of the specified logic channel of condition A.	5-124
:TRIGger:ATRigger:CHANnel<x>:SCHannel<x>:BIT<x>	Sets or queries the state of the specified bit of the specified sub channel of the specified logic channel of condition A.	5-124
:TRIGger:ATRigger:CHANnel<x>[:CONDition]	Sets or queries the state of the specified channel of the specified logic channel of condition A.	5-124
:TRIGger:ATRigger:CHANnel<x>:SCHannel<x>[:CONDition]	Sets or queries the state of the specified sub channel of the specified logic channel of condition A.	5-125
:TRIGger:ATRigger:CONDition	Sets or queries the achievement condition of condition A.	5-125
:TRIGger:BGTime?	Queries all B>Time trigger settings.	5-125
:TRIGger:BGTime:TIME	Sets or queries the pulse width for B>Time triggers.	5-125
:TRIGger:BLTime?	Queries all B<Time trigger settings.	5-125
:TRIGger:BLTime:TIME	Sets or queries the pulse width for B<Time triggers.	5-125
:TRIGger:BTOut?	Queries all B TimeOut trigger settings.	5-125
:TRIGger:BTOut:TIME	Sets or queries the pulse width for B TimeOut triggers.	5-125
:TRIGger:BBETween?	Queries all B Between trigger settings.	5-125
:TRIGger:BBETween:TIME<x>	Sets or queries a pulse width for B Between triggers.	5-125
:TRIGger:BTRigger?	Queries all condition B settings.	5-125
:TRIGger:BTRigger:CHANnel<x>:BIT<x>	Sets or queries the state of the specified bit of the specified logic channel of condition B.	5-126
:TRIGger:BTRigger:CHANnel<x>:SCHannel<x>:BIT<x>	Sets or queries the state of the specified bit of the specified sub channel of the specified logic channel of condition B.	5-126
:TRIGger:BTRigger:CHANnel<x>[:CONDition]	Sets or queries the state of the specified channel of the specified logic channel of condition B.	5-126
:TRIGger:BTRigger:CHANnel<x>:SCHannel<x>[:CONDition]	Sets or queries the state of the specified sub channel of the specified logic channel of condition B.	5-126
:TRIGger:BTRigger:CONDition	Sets or queries the achievement condition of condition B.	5-126
:TRIGger:DELay	Sets or queries the delay.	5-126
:TRIGger:EOA?	Queries all EdgeOnA settings.	5-126
:TRIGger:EOA:CHANnel<x>:BIT<x>	Sets or queries the state of the specified bit of the specified logic channel for EdgeOnA triggers.	5-127
:TRIGger:EOA:CHANnel<x>:SCHannel<x>:BIT<x>	Sets or queries the state of the specified bit of the specified sub channel of the specified logic channel for EdgeOnA triggers.	5-127
:TRIGger:EOA:CHANnel<x>[:CONDition]	Sets or queries the state of the specified channel of the specified logic channel for EdgeOnA triggers.	5-127
:TRIGger:EOA:CHANnel<x>:SCHannel<x>[:CONDition]	Sets or queries the state of the specified sub channel of the specified logic channel for EdgeOnA triggers.	5-127
:TRIGger:EOA:CONDition	Sets or queries the achievement condition for EdgeOnA triggers.	5-127
:TRIGger:HOLDoff?	Queries all hold-off settings.	5-127
:TRIGger:HOLDoff:TIME	Sets or queries the hold-off time.	5-127
:TRIGger:MODE	Sets or queries the trigger mode.	5-127
:TRIGger:OR?	Queries all OR trigger settings.	5-128

Command	Function	Page
:TRIGger:OR:CHANnel<x>:BIT<x>	Sets or queries the state of the specified bit of the specified logic channel for OR triggers.	5-128
:TRIGger:OR:CHANnel<x>:SCHannel<x>:BIT<x>	Sets or queries the state of the specified bit of the specified sub channel of the specified logic channel for OR triggers.	5-128
:TRIGger:OR:CHANnel<x>[:CONDition]	Sets or queries the state of the specified channel of the specified logic channel for OR triggers.	5-128
:TRIGger:OR:CHANnel<x>:SCHannel<x>[:CONDition]	Sets or queries the state of the specified sub channel of the specified logic channel for OR triggers.	5-128
:TRIGger:OR:EXTernal:TYPE	Sets or queries the type of external trigger that is used with OR triggers.	5-128
:TRIGger:OUT?	Queries all trigger output settings.	5-128
:TRIGger:OUT:TIME	Sets or queries the H-pulse interval that is used when the trigger output terminal output type is 1.	5-128
:TRIGger:OUT:TYPE	Sets or queries the trigger output terminal output type.	5-128
:TRIGger:POSition	Sets or queries the trigger position.	5-128
:TRIGger:SIMPle?	Queries all simple trigger settings.	5-129
:TRIGger[:SIMPle]:EXTernal:SLOPe	Sets or queries the external trigger slope.	5-129
:TRIGger[:SIMPle]:HYSTeresis	Sets or queries the hysteresis for simple triggers.	5-129
:TRIGger[:SIMPle]:LEVel	Sets or queries the trigger level of the channel specified by :TRIGger:SIMPle:SOURce for simple triggers .	5-129
:TRIGger[:SIMPle]:LOGic?	Queries all simple trigger (logic trigger) settings.	5-129
:TRIGger[:SIMPle]:LOGic:BIT	Sets or queries the bit that sets the trigger slope.	5-129
:TRIGger[:SIMPle]:LOGic:SLOPe	Sets or queries the slope for logic triggers.	5-129
:TRIGger[:SIMPle]:SLOPe	Sets or queries the trigger slope of the channel specified by :TRIGger:SIMPle:SOURce for simple triggers .	5-129
:TRIGger:SIMPle:SOURce	Sets or queries the source for simple triggers.	5-129
:TRIGger:SOURce?	Queries all the settings of the trigger source for enhanced triggers.	5-129
:TRIGger:SOURce:CHANnel<x>[:SCHannel<x>]?	Queries all the settings of the specified channel's trigger source for enhanced triggers.	5-130
:TRIGger:SOURce:CHANnel<x>[:SCHannel<x>]:CENTer	Sets or queries the window trigger center for enhanced triggers.	5-130
:TRIGger:SOURce:CHANnel<x>[:SCHannel<x>]:HYSTeresis	Sets or queries the specified channel's hysteresis for enhanced triggers.	5-130
:TRIGger:SOURce:CHANnel<x>[:SCHannel<x>]:LEVel	Sets or queries the specified channel's level for enhanced triggers.	5-130
:TRIGger:SOURce:CHANnel<x>[:SCHannel<x>]:WIDTh	Sets or queries the window trigger width for enhanced triggers.	5-130
:TRIGger:SOURce:CHANnel<x>:WWIDth	Sets or queries the width for wave window triggers.	5-130
:TRIGger:TGTime?	Queries all T>Time trigger settings.	5-130
:TRIGger:TGTime:TIME	Sets or queries the pulse width for T>Time triggers.	5-131
:TRIGger:TIMer?	Queries all time trigger settings.	5-131
:TRIGger:TIMer:DATE	Sets or queries the date for time triggers.	5-131
:TRIGger:TIMer:INTerval	Sets or queries the trigger interval for time triggers.	5-131
:TRIGger:TIMer:TIME	Sets or queries the time for time triggers.	5-131
:TRIGger:TITime?	Queries all T1<T<T2 trigger settings.	5-131
:TRIGger:TITime:TIME<x>	Sets or queries the pulse width for T1<T<T2 triggers.	5-131
:TRIGger:TLTime?	Queries all T<Time trigger settings.	5-131
:TRIGger:TLTime:TIME	Sets or queries the pulse width for T<Time triggers.	5-131
:TRIGger:TOTime?	Queries all T<T1,T2<T trigger settings.	5-131
:TRIGger:TOTime:TIME<x>	Sets or queries the pulse width for T<T1,T2<T triggers.	5-131
:TRIGger:TYPE	Sets or queries the trigger type.	5-132
:TRIGger:WWINdow?	Queries all wave window trigger settings.	5-132
:TRIGger:WWINdow:CHANnel<x>	Sets or queries the state of the specified channel for wave window triggers.	5-132
:TRIGger:WWINdow:FREQuency	Sets or queries the cycle frequency for wave window triggers.	5-132
:TRIGger:WWINdow:REFCycle	Sets or queries the reference cycle for wave window triggers.	5-132
:TRIGger:WWINdow:SYNC?	Queries all synchronization channel settings for wave window triggers.	5-132
:TRIGger:WWINdow:SYNC:HYSTeresis	Sets or queries the hysteresis of the synchronization channel for wave window triggers.	5-132
:TRIGger:WWINdow:SYNC:LEVel	Sets or queries the level of the synchronization channel for wave window triggers.	5-132
:TRIGger:WWINdow:SYNC:TRACe	Sets or queries the synchronization channel for wave window triggers.	5-132

Command	Function	Page
WAVeform Group		
:WAVeform?	Queries all waveform data output settings.	5-133
:WAVeform:BITS?	Queries the bit length of the waveform data specified by the :WAVeform:TRACe command.	5-133
:WAVeform:BYTeorder	Sets or queries the transmission byte order for data formats that are 2 bytes or longer.	5-133
:WAVeform:CAPTure?	Queries all capture data output settings.	5-133
:WAVeform:CAPTure:END	Sets or queries the end point to use when capturing waveform data.	5-133
:WAVeform:CAPTure:LENGth?	Queries the number of data points in the captured waveform.	5-133
:WAVeform:CAPTure:RECord	Sets or queries the record number to use when capturing waveform data.	5-133
:WAVeform:CAPTure:RECord? MAXimum	Queries the largest record number of the captured waveform.	5-133
:WAVeform:CAPTure:RECord? MINimum	Queries the smallest record number of the captured waveform.	5-133
:WAVeform:CAPTure:SEND?	Queries specified capture waveform data.	5-133
:WAVeform:CAPTure:SRATe?	Queries the sample rate of the capture waveform specified by the :WAVeform:TRACe command.	5-134
:WAVeform:CAPTure:STARt	Sets or queries the start point to use when capturing waveform data.	5-134
:WAVeform:CAPTure:TRIGger?	Queries the trigger position (represented as a number of points) in the captured waveform.	5-134
:WAVeform:DATaselect	Queries whether to query the waveform specified by the :WAVeform:TRACe command using ACQ data or PP data.	5-134
:WAVeform:END	Sets or queries the end data point in the waveform specified by the :WAVeform:TRACe command (the main waveform).	5-134
:WAVeform:FORMat	Sets or queries the transmission data format.	5-134
:WAVeform:LENGth?	Queries the total number of data points in the waveform specified by the :WAVeform:TRACe command (the main waveform).	5-134
:WAVeform:MODule?	Queries the module of the waveform specified by the :WAVeform:TRACe command.	5-134
:WAVeform:OFFSet?	Queries the offset value used to convert the waveform data specified by the :WAVeform:TRACe command to physical values.	5-135
:WAVeform:RANGe?	Queries the measurement range used to convert the waveform data specified by the :WAVeform:TRACe command to physical values.	5-135
:WAVeform:RECord	Sets or queries the main waveform record number that WAVeform commands will be applied to.	5-135
:WAVeform:RECord? MINimum	Queries the minimum record number in the history memory (for the main waveform).	5-135
:WAVeform:SEND?	Queries the waveform data specified by the :WAVeform:TRACe command (the main waveform data).	5-135
:WAVeform:SIGN?	Queries whether signs are included in the block data of the source waveform data specified by :WAVeform:TRACe when the data is queried.	5-136
:WAVeform:SRATe? (Sample RATE)	Queries the sample rate of the waveform specified by the :WAVeform:TRACe command.	5-136
:WAVeform:STARt	Sets or queries the start data point in the waveform specified by the :WAVeform:TRACe command (the main waveform).	5-136
:WAVeform:TRACe	Sets or queries the waveform that WAVeform commands will be applied to.	5-136
:WAVeform:TRIGger?	Queries the trigger position of the record specified by the :WAVeform:RECord command.	5-136
:WAVeform:TYPE?	Queries the acquisition mode of the source waveform.	5-136
XY Group		
:XY?	Queries all XY settings.	5-137
:XY:CDISplay (Combine Display)	Sets or queries whether to combine the displays of Window1 and Window2 on the XY waveform display.	5-137
:XY:DOTConnect	Sets or queries whether dot connect is ON or OFF for XY waveforms.	5-137
:XY:DECimation	Sets or queries the number of dots (after decimation) that XY waveforms use.	5-137
:XY:MARKer	Sets or queries whether XY pen markers are on.	5-137
:XY:TCLear (Trace Clear On Start)	Sets or queries whether the XY trace-clear-on-start feature is on.	5-137

Command	Function	Page
:XY:WINDow<x>?	Queries all settings related to the specified XY window.	5-137
:XY:WINDow<x>:TRANge	Sets or queries the T-Y waveform range to display in the XY window.	5-137
:XY:WINDow<x>:MODE	Sets or queries whether the specified XY window is displayed.	5-137
:XY:WAVeform<x1>?	Queries all settings related to the specified XY waveform.	5-137
:XY:WAVeform<x1>:DISPlay	Sets or queries whether the specified XY waveform is displayed.	5-137
:XY:WAVeform<x1>:XTRace	Sets or queries the channel that is assigned to the specified XY waveform's X axis.	5-138
:XY:WAVeform<x1>:YTRace	Sets or queries the channel that is assigned to the specified XY waveform's Y axis.	5-138

ZOOM Group

:ZOOM?	Queries all zoom settings.	5-139
:ZOOM:ASCRoll?	Queries all auto scroll settings.	5-139
:ZOOM:ASCRoll:JUMP	Moves the center position of the zoom box to the left or right edge of the main window.	5-139
:ZOOM:ASCRoll:SPEed	Sets or queries the auto scroll speed of the zoom box.	5-139
:ZOOM:ASCRoll:STARt	Starts auto scrolling.	5-139
:ZOOM:ASCRoll:STOP	Stops auto scrolling.	5-139
:ZOOM:ASCRoll:TARGet	Sets or queries the zoom window that will be auto scrolled.	5-139
:ZOOM:FORMat<x>	Sets or queries the display format of the specified zoom waveform.	5-139
:ZOOM:GROup<x1>?	Queries all settings related to the specified group display of zoom waveforms.	5-139
:ZOOM:GROup<x1>:TRACe<x2>	Sets or queries whether the specified source waveform of the specified group display of zoom waveforms is displayed.	5-139
:ZOOM:MAG<x>	Sets or queries the horizontal magnification of the specified zoom waveform.	5-140
:ZOOM:MAIN	Sets or queries the proportion of the main waveform display area that is used when zooming waveforms.	5-140
:ZOOM:MODE<x>	Sets or queries whether the specified zoom waveform is displayed.	5-140
:ZOOM:MOVE	Sets or queries whether the movement of the zoom position to the latest position is enabled or disabled.	5-140
:ZOOM:POSition<x>	Sets or queries the zoom position of the specified zoom waveform.	5-140
:ZOOM:TDIV<x>	Sets or queries the T/DIV value of the specified zoom waveform.	5-140
:ZOOM:WLAYout (Window Layout)	Sets or queries the window layout that is used when waveforms are zoomed.	5-140
:ZOOM:Z2Target	Sets or queries the source window of Z2 when both Z1 and Z2 are displayed.	5-140

Common Command Group

*CAL?	Executes calibration and queries the result.	5-141
*CLS	Clears the standard event register, extended event register, and error queue.	5-141
*ESE	Sets or queries the standard event enable register.	5-141
*ESR?	Queries and clears the standard event register.	5-141
*IDN?	Queries the DL850/DL850V model.	5-141
*OPC	Sets bit 0 (the OPC bit) of the standard event register to 1 upon the completion of the specified overlap command.	5-141
*OPC?	Returns ASCII code 1 when the specified overlap command is completed.	5-141
*OPT?	Queries the installed options.	5-142
*RST	Initializes settings.	5-142
*SRE	Sets or queries the service request enable register value.	5-142
*STB?	Queries the status byte register value.	5-142
*TST?	Performs a self-test and queries the result. The self-test consists of tests of each kind of internal memory.	5-142
*WAI	Holds the execution of the subsequent command until the specified overlap command is completed.	5-142

5.2 ACQuire Group

The commands in this group deal with waveform acquisition. You can make the same settings and queries that you can by using the ACQUIRE key on the front panel.

:ACQuire?

Function Queries all waveform acquisition settings.
Syntax :ACQuire?

:ACQuire:AVERage?

Function Queries all averaging settings.
Syntax :ACQuire:AVERage?

:ACQuire:AVERage:COUNt

Function Sets or queries the number of waveform acquisitions to perform during averaging.
Syntax :ACQuire:AVERage:COUNt {<NRf>|INFinity}
<NRf> = 2 to 65536 (in 2^n steps)
Example :ACQUIRE:AVERAGE:COUNT INFINITY
:ACQUIRE:AVERAGE:COUNT? ->
:ACQUIRE:AVERAGE:COUNT INFINITY

:ACQuire:AVERage:EWEight (Exponent Weight)

Function Sets or queries the attenuation constant of exponential averaging.
Syntax :ACQuire:AVERage:EWEight {<NRf>}
<NRf> = 2 to 256 (in 2^n steps)
Example :ACQUIRE:AVERAGE:EWEIGHT 16
:ACQUIRE:AVERAGE:EWEIGHT? ->
:ACQUIRE:AVERAGE:EWEIGHT 16

:ACQuire:CLOCk

Function Sets or queries the time base (internal or external clock).
Syntax :ACQuire:CLOCk {INTernal|EXTernal}
Example :ACQUIRE:CLOCK INTERNAL
:ACQUIRE:CLOCK? ->
:ACQUIRE:CLOCK INTERNAL

:ACQuire:COUNt

Function Sets or queries the number of waveform acquisitions to perform in Normal mode.
Syntax :ACQuire:COUNt {<NRf>|INFinity}
<NRf> = 1 to 65536
Example :ACQUIRE:COUNT INFINITY
:ACQUIRE:COUNT? ->
:ACQUIRE:COUNT INFINITY

:ACQuire:MODE

Function Sets or queries the waveform acquisition mode.
Syntax :ACQuire:MODE {AVERage|BAVerage|ENVelope|NORMal}
Example :ACQUIRE:MODE NORMAL
:ACQUIRE:MODE? ->
:ACQUIRE:MODE NORMAL

:ACQuire:PROTate

Function Sets or queries the pulse/rotate setting to use during external clock input.
Syntax :ACQuire:PROTate {<NRf>}
<NRf> = 1 to 24000
Example :ACQUIRE:PROTATE 100
:ACQUIRE:PROTATE? ->
:ACQUIRE:PROTATE 100

:ACQuire:RLENgth

Function Sets or queries the record length.
Syntax :ACQuire:RLENgth {<NRf>}
<NRf> = 1000 to 2000000000
Step: 1000, 2500, 5000, 10000, 25000, 50000, 100000, 250000, 500000, 1000000, 2500000, 5000000, 10000000, 25000000, 50000000, 100000000, 250000000, 500000000, 1000000000, 2000000000 (only possible during hard disk recording), 5000000000, 10000000000, 20000000000, 50000000000
Example :ACQUIRE:RLENGTH 10000
:ACQUIRE:RLENGTH? ->
:ACQUIRE:RLENGTH 10000

:ACQuire:RTOut?

Function Queries all hard-disk-recording settings.
Syntax :ACQuire:RTOut?

:ACQuire:RTOut:ANAMing

Function Sets or queries the hard-disk-recording auto naming feature.

Syntax `:ACQuire:RTOut:ANAMing {DATE|NUMBering}`

Example
```
:ACQUIRE:RTOUT:ANAMING DATE
:ACQUIRE:RTOUT:ANAMING? ->
:ACQUIRE:RTOUT:ANAMING DATE
```

:ACQuire:RTOut:COMMent

Function Sets or queries the hard-disk-recording comment.

Syntax
```
:ACQuire:RTOut:COMMent <String>
<String> = Up to 120 characters
```

Example
```
:ACQUIRE:RTOUT:COMMENT "ABC"
:ACQUIRE:RTOUT:COMMENT? ->
:ACQUIRE:RTOUT:COMMENT "ABC"
```

:ACQuire:RTOut:FILename

Function Sets or queries the hard-disk-recording file name.

Syntax
```
:ACQuire:RTOut:FILename <String>
<String> = Up to 16 characters
```

Example
```
:ACQUIRE:RTOUT:FILENAME "ABC"
:ACQUIRE:RTOUT:FILENAME? ->
:ACQUIRE:RTOUT:FILENAME "ABC"
```

:ACQuire:RTOut:MODE

Function Sets or queries whether hard disk recording is enabled.

Syntax `:ACQuire:RTOut:MODE {HD|OFF}`

Example
```
:ACQUIRE:RTOUT:MODE OFF
:ACQUIRE:RTOUT:MODE? ->
:ACQUIRE:RTOUT:MODE OFF
```

:ACQuire:RTOut:DIVide:MODE

Function Sets or queries whether divided recording is enabled for hard disk recording.

Syntax
```
:ACQuire:RTOut:DIVide:
MODE {<Boolean>}
```

Example
```
:ACQUIRE:RTOUT:DIVIDE:MODE ON
:ACQUIRE:RTOUT:DIVIDE:MODE? ->
:ACQUIRE:RTOUT:DIVIDE:MODE 1
```

:ACQuire:RTOut:DIVide:NUMBer

Function Sets or queries the number of divisions when divided recording is performed during hard disk recording.

Syntax
```
:ACQuire:RTOut:DIVide:NUMBer {<NRf>}
<NRf> = 10, 20, 50, 100
```

Example
```
:ACQUIRE:RTOUT:DIVIDE:NUMBER 10
:ACQUIRE:RTOUT:DIVIDE:NUMBER? ->
:ACQUIRE:RTOUT:DIVIDE:NUMBER 10
```

5.3 ASETup Group

The commands in this group deal with auto setup. You can perform the same operations and make the same settings and queries that you can by using the SETUP key on the front panel.

:ASETup?

Function Queries all auto setup settings.
Syntax `:ASETup?`

:ASETup:EXECute

Function Executes auto setup.
Syntax `:ASETup:EXECute`
Description Auto setup cannot be executed during hard disk recording.

:ASETup:TARGet

Function Sets or queries the channel to perform auto setup on.
Syntax
```
ASETup:TARGet {ALL|<NRf>[,<NRf>]}
<NRf> = 1 to 16
[,<NRf>] = If the channel has sub
channels, this is the range of sub
channel numbers.
```
Example
```
:ASETUP:TARGET ALL
:ASETUP:TARGET? ->
:ASETUP:TARGET ALL
```
Description An error will occur if you specify a channel that does not have a module installed in it.
An error will occur if you specify a sub channel that does not exist.

:ASETup:UNDO

Function Undoes auto setup.
Syntax `:ASETup:UNDO`

5.4 CALibrate Group

The commands in this group deal with calibration. You can perform the same operations and make the same settings and queries that you can by using the SHIFT+SETUP keys on the front panel.

:CALibrate?

Function Queries all calibration settings.
Syntax `:CALibrate?`

:CALibrate[:EXECute]

Function Executes calibration.
This command is different from the common command *CAL? in that this command does not return the results when the calibration is completed.
This is an overlap command.
Syntax `CAL:EXEC`

:CALibrate:MODE

Function Sets or queries the auto calibration mode.
Syntax `CALibrate:MODE {AUTO|OFF}`
Example
```
:CALIBRATE:MODE AUTO
:CALIBRATE:MODE? ->
:CALIBRATE:MODE AUTO
```

5.5 CAPTure Group

The commands in this group deal with the dual capture feature. You can make the same settings and queries that you can by using the SHIFT+ACQUIRE keys on the front panel.

:CAPTure?
Function Queries all dual capture settings.
Syntax :CAPTure?

:CAPTure:ACTion?
Function Queries all dual-capture-action settings.
Syntax :CAPTure:ACTion?

:CAPTure:ACTion:BUZZer
Function Sets or queries whether a beep is sounded as one of the capture actions.
Syntax :CAPTure:ACTion:BUZZer {<Boolean>}
Example :CAPTURE:ACTION:BUZZER ON
:CAPTURE:ACTION:BUZZER? ->
:CAPTURE:ACTION:BUZZER 1
Description This setting is shared with the corresponding buzzer settings of other actions.

:CAPTure:ACTion:MAIL?
Function Queries all dual-capture-action e-mail transmission settings.
Syntax :CAPTure:ACTion:MAIL?

:CAPTure:ACTion:MAIL:COUNt
Function Sets or queries the upper limit of e-mail transmissions to perform as a capture action.
Syntax :CAPTure:ACTion:MAIL:COUNt {<NRf>}
<NRf> = 1 to 1000
Example :CAPTURE:ACTION:MAIL:COUNT 100
:CAPTURE:ACTION:MAIL:COUNT? -> :
CAPTURE:ACTION:MAIL:COUNT 100
Description This setting is shared with the corresponding e-mail transmission settings of other actions.

:CAPTure:ACTion:MAIL:MODE
Function Sets or queries whether e-mail is transmitted as a capture action.
Syntax :CAPTure:ACTion:MAIL:
MODE {<Boolean>}
Example :CAPTURE:ACTION:MAIL:MODE ON
:CAPTURE:ACTION:MAIL:MODE? ->
:CAPTURE:ACTION:MAIL:MODE 1
Description This setting is shared with the corresponding e-mail transmission settings of other actions

:CAPTure:ACTion:SAVE:ANAMing
Function Sets or queries the auto file naming method.
Syntax :CAPTure:ACTion:SAVE:ANAMing {DATE|
NUMBering}
Example :CAPTURE:ACTION:SAVE:
ANAMING NUMBERING
:CAPTURE:ACTION:SAVE:ANAMING? ->
:CAPTURE:ACTION:SAVE:
ANAMING NUMBERING
Description This setting is shared with the auto-naming settings of other actions.

:CAPTure:ACTion:SAVE?
Function Queries all dual-capture-action data save settings.
Syntax :CAPTure:ACTion:SAVE?

:CAPTure:ACTion:SAVE[:MODE]
Function Sets or queries whether captured data is saved to a storage medium as a capture action.
Syntax :CAPTure:ACTion:
SAVE[:MODE] {<Boolean>}
:CAPTure:ACTion:SAVE:MODE?
Example :CAPTURE:ACTION:SAVE:MODE ON
:CAPTURE:ACTION:SAVE:MODE? ->
:CAPTURE:ACTION:SAVE:MODE 1
Description This setting is shared with the corresponding data-save settings of other actions.

:CAPTure:ACTion:SAVE:NAME
Function Sets or queries the file name that is used when captured data is saved to a storage medium.
Syntax :CAPTure:ACTion:SAVE:NAME <Filename>
Example :CAPTURE:ACTION:SAVE:NAME "ABC"
:CAPTURE:ACTION:SAVE:NAME? ->
:CAPTURE:ACTION:SAVE:NAME "ABC"
Description This setting is shared with other file name settings.

:CAPTure:ACTion:SAVE:CDIRectory
Function Sets the current directory on the storage medium where the screen capture is saved to as a capture action.
Syntax :CAPTure:ACTion:SAVE:
CDIRectory {<String>}
Example :CAPTURE:ACTION:SAVE:
CDIRECTORY "ABCD"

:CAPTure:ACTion:SAVE:TYPE

Function Sets or queries the data format that is used when captured data is saved to a storage medium.

Syntax
```
:CAPTure:ACTion:SAVE:TYPE {ASCii|
BINary|FLOat}
:CAPTure:ACTion:SAVE:TYPE?
```

Example
```
:CAPTURE:ACTION:SAVE:TYPE BINARY
:CAPTURE:ACTION:SAVE:TYPE? ->
:CAPTURE:ACTION:SAVE:TYPE BINARY
```

Description This setting is shared with the corresponding data format settings of other actions.

:CAPTure:BITDisp

Function Sets or queries whether captured event waveforms are displayed.

Syntax
```
:CAPTure:BITDisp {<Boolean>}
```

Example
```
:CAPTURE:BITDISP ON
:CAPTURE:BITDISP? ->
:CAPTURE:BITDISP 1
```

:CAPTure:CAPNum? MAXimum

Function Queries the largest number of the captured waveforms.

Syntax
```
:CAPTure:CAPNum? MAXimum
```

Description This command is valid when waveforms are acquired with the capture mode set to On Start.

:CAPTure:CAPNum? MINimum

Function Queries the smallest number of the captured waveforms.

Syntax
```
:CAPTure:CAPNum? MINimum
```

Description This command is valid when waveforms are acquired with the capture mode set to Auto.

:CAPTure:FORMat

Function Sets or queries the display format of the capture window.

Syntax
```
:CAPTure:FORMat {MAIN|<NRf>}
:CAPTure:FORMat?
<NRf> = 1, 2, 3, 4, 6, 8, 12, 16
```

Example
```
:CAPTURE:FORMAT MAIN
:CAPTURE:FORMAT? ->
:CAPTURE:FORMAT MAIN
```

:CAPTure:GROup<x1>?

Function Queries all group display settings of the capture window.

Syntax
```
:CAPTure:GROup<x1>?
```

:CAPTure:GROup<x1>:TRACe<x2>

Function Sets or queries the displayed waveform in the specified group display of the capture window.

Syntax
```
:CAPTure:GROup<x1>:
TRACe<x2> {<Boolean>}
:CAPTure:GROup<x1>:TRACe<x2>?
<x1> = 1 to 4
<x2> = 1 to 64
```

Example
```
:CAPTURE:GROUP1:TRACE1 1
:CAPTURE:GROUP1:TRACE1? ->
:CAPTURE:GROUP1:TRACE1 1
```

:CAPTure:MCMode (Main Capture Mode)

Function Sets or queries the main capture mode during dual capture.

Syntax
```
:CAPTure:MCMode {AUTO|ONSTart}
:CAPTure:MCMode?
```

Example
```
:CAPTURE:MCMODE AUTO
:CAPTURE:MCMODE? ->
:CAPTURE:MCMODE AUTO
```

:CAPTure:MODE

Function Sets or queries whether the dual capture mode is enabled.

Syntax
```
:CAPTure:MODE {<Boolean>}
:CAPTure:MODE?
```

Example
```
:CAPTURE:MODE ON
:CAPTURE:MODE? -> :CAPTURE:MODE 1
```

:CAPTure:RLENgth

Function Sets or queries the dual capture length.

Syntax
```
:CAPTure:RLENgth {<NRf>}
:CAPTure:RLENgth?
<NRf> = 5000 to 500000
Step: 5000, 10000, 25000, 50000,
100000, 250000, 500000
```

Example
```
:CAPTURE:RLENGTH 10000
:CAPTURE:RLENGTH? ->
:CAPTURE:RLENGTH 10000
```

:CAPTure:TDIV

Function Sets or queries the dual capture T/DIV setting.

Syntax
```
:CAPTure:TDIV{<Time>}
:CAPTure:TDIV?
<Time> = 1us to 60s
```

Example
```
:CAPTURE:TDIV 100.0E-06
:CAPTURE:TDIV? -> :CAPTURE:
TDIV 100.0E-06
```

:CAPTure:WINDow?

Function Queries all dual-capture-display-window settings.

Syntax
```
:CAPTure:WINDow?
```

:CAPTure:WINDow:MAG

Function Sets or queries the dual-capture-window zoom factor as a time.

Syntax
```
:CAPTure:WINDow:MAG {<Time>}
:CAPTure:WINDow:MAG?
```

Example
```
:CAPTURE:WINDOW:MAG 100.0E-06
:CAPTURE:WINDOW:MAG? ->
:CAPTURE:WINDOW:MAG 100.0E-06
<NRf> = 1us to 60s
```

:CAPTure:WINDow:MODE

Function Sets or queries whether the dual capture window is turned on.

Syntax
```
:CAPTure:WINDow<x>:MODE <Boolean>
:CAPTure:WINDow<x>:MODE?
```

Example
```
:CAPTURE:WINDOW:MODE ON
:CAPTURE:WINDOW:MODE? ->
:CAPTURE:WINDOW:MODE 1
```

:CAPTure:WINDow:CAPNum

Function Sets or queries the number of the waveform that is displayed in the dual capture window.

Syntax
```
:CAPTure:WINDow:CAPNum {<NRf>|
MAXimum|MINimum}
:CAPTure:WINDow:CAPNum?
<NRf> = -5000 to 5000
```

Example
```
:CAPTURE:WINDOW:CAPNUM -1
:CAPTURE:WINDOW:CAPNUM? ->
:CAPTURE:WINDOW:CAPNUM -1
```

Description When the capture mode is set to Auto:Current, –1, –2, ...Current: The latest waveform; –1: The waveform that was captured before the Current waveform; –2: The waveform that was captured two waveforms before the Current waveform. If you specify MAXimum, the Current waveform is specified. If you specify MINimum, the number of the oldest waveform is specified.
When the capture mode is set to On Start: Current, 1, 2, ...Current: The latest waveform; 1: The oldest waveform: 2: The second oldest waveform, ... the waveform that was captured before the Current waveform. If you specify MAXimum, the waveform that was captured before the Current waveform is specified. If you specify MINimum, the number of the oldest waveform is specified.

:CAPTure:WINDow:POSition

Function Sets or queries the dual-capture-window horizontal position.

Syntax
```
:CAPTure:WINDow<x>:POSition {<NRf>}
:CAPTure:WINDow<x>:POSition?
<NRf> = -5 to 5div
```

Example
```
:CAPTURE:WINDOW:POSITION 0
:CAPTURE:WINDOW:POSITION? ->
:CAPTURE:WINDOW:
POSITION 0.000000000000
```

5.6 CHANnel Group

The commands in this group deal with a channel's vertical axis. You can make the same settings and queries that you can by using the VERTICAL group keys (CH and V/DIV keys) on the front panel.

:CHANnel<x>?

Function Queries all vertical axis settings of a channel.

Syntax
```
:CHANnel<x>?
```

:CHANnel<x>:ACCL?

Function Queries all the settings of an installed acceleration/voltage module.

```
<x> = 1 to 16
```

:CHANnel<x>:ACCL:BIAS

Function Sets or queries whether the bias current supply to the acceleration sensors of an installed acceleration/voltage module is on.

Syntax
```
:CHANnel<x>:ACCL:BIAS {<Boolean>}
:CHANnel<x>:ACCL:BIAS?
<x> = 1 to 16
```

Example
```
:CHANNEL1:ACCL:BIAS ON
:CHANNEL1:ACCL:BIAS? ->
:CHANNEL1:ACCL:BIAS 1
```

:CHANnel<x>:ACCL:BWIDth

Function Sets or queries the filter when the input coupling of an installed acceleration/voltage module is set to acceleration.

Syntax
```
:CHANnel<x>:ACCL:BWIDth {FULL|AUTO|
<Frequency>
:CHANnel<x>:ACCL:BWIDth?
<x> = 1 to 16
<Frequency> = 4kHz, 400Hz, 40Hz
```

Example
```
:CHANNEL1:ACCL:BWIDTH FULL
:CHANNEL1:ACCL:BWIDTH? ->
:CHANNEL1:ACCL:BWIDTH FULL
```

:CHANnel<x>:ACCL:COUPling

Function Sets or queries the input coupling setting of an installed acceleration/voltage module.

Syntax
```
:CHANnel<x>:ACCL:COUPling {AC|DC|
ACCL|GND}
:CHANnel<x>:ACCL:COUPling?
<x> = 1 to 16
```

Example
```
:CHANNEL1:ACCL:COUPLING ACCL
:CHANNEL1:ACCL:COUPLING? ->
:CHANNEL1:ACCL:COUPLING ACCL
```

:CHANnel<x>:ACCL:GAIN

Function Sets or queries the gain when the input coupling of an installed acceleration/voltage module is set to acceleration.

Syntax
```
:CHANnel<x>:ACCL:GAIN {<NRf>}
:CHANnel<x>:ACCL:GAIN?
<x> = 1 to 16
<NRf> = 0.1, 0.2, 0.5, 1, 2, 5, 10,
20, 50, 100
```

Example
```
:CHANNEL1:ACCL:GAIN 1.0
:CHANNEL1:ACCL:GAIN? ->
:CHANNEL1:ACCL:GAIN 1.0
```

:CHANnel<x>:ACCL:POSition

Function Sets or queries the vertical position when the input coupling of an installed acceleration/voltage module is set to acceleration.

Syntax
```
:CHANnel<x>:ACCL:POSition {<NRf>}
:CHANnel<x>:ACCL:POSition?
<x> = 1 to 16
<NRf> = -5.00 to 5.00 (div; in 0.01
div steps)
```

Example
```
:CHANNEL1:ACCL:POSITION 0.00
:CHANNEL1:ACCL:POSITION? ->
:CHANNEL1:ACCL:POSITION 0.00
```

:CHANnel<x>:ACCL:SCALe

Function Sets or queries the upper and lower limits of the screen when the input coupling of an installed acceleration/voltage module is set to acceleration.

Syntax
```
:CHANnel<x>:ACCL:SCALe {<NRf>,<NRf>}
:CHANnel<x>:ACCL:SCALe?
<x> = 1 to 16
<NRf> = 1000000 to -1000000
```

Example
```
:CHANNEL1:ACCL:SCALE
5000.00,-5000.00
:CHANNEL1:ACCL:SCALE? -> :CHANNEL1
:ACCL:SCALE 5000.00,-5000.00
```

:CHANnel<x>:ACCL:SENSitivity

Function Sets or queries the sensitivity when the input coupling of an installed acceleration/voltage module is set to acceleration.

Syntax
```
:CHANnel<x>:ACCL:SENSitivity {<NRf>}
:CHANnel<x>:ACCL:SENSitivity?
<x> = 1 to 16
<NRf> = 0.1 to 2000
```

Example
```
:CHANNEL1:ACCL:SENSITIVITY 1.00
:CHANNEL1:ACCL:SENSITIVITY? ->
:CHANNEL1:ACCL:SENSITIVITY 1.00
```

:CHANnel<x>:ACCL:UNIT

Function Sets or queries the unit string when the input coupling of an installed acceleration/voltage module is set to acceleration.

Syntax
```
:CHANnel<x>:ACCL:UNIT {<String>}
:CHANnel<x>:ACCL:UNIT?
<x> = 1 to 16
<String> = Up to 4 characters
```

Example
```
:CHANNEL1:ACCL:UNIT "m/s2"
:CHANNEL1:ACCL:UNIT? ->
:CHANNEL1:ACCL:UNIT "m/s2"
```

:CHANnel<x>:ACCL:VARiable

Function Sets or queries the scale setup mode when the input coupling of an installed acceleration/voltage module is set to acceleration.

Syntax
```
:CHANnel<x>:ACCL:
VARiable {<Boolean>}
:CHANnel<x>:ACCL:VARiable?
<x> = 1 to 16
```

Example
```
:CHANNEL1:ACCL:VARIABLE 0
:CHANNEL1:ACCL:VARIABLE? ->
:CHANNEL1:ACCL:VARIABLE 0
```

:CHANnel<x>:ACCL:ZOOM

Function Sets or queries the vertical zoom factor when the input coupling of an installed acceleration/voltage module is set to acceleration.

Syntax
```
:CHANnel<x>:ACCL:ZOOM {<NRf>}
:CHANnel<x>:ACCL:ZOOM?
<x> = 1 to 16
<NRf> = 0.5, 0.556, 0.625, 0.667,
0.714, 0.8, 0.833, 1, 1.11, 1.25,
1.33, 1.43,
1.67, 2, 2.22, 2.5, 3.33, 4, 5, 6.67,
8, 10, 12.5, 16.7, 20, 25, 40, 50
```

Example
```
:CHANNEL1:ACCL:ZOOM 1.000
:CHANNEL1:ACCL:ZOOM? -> :CHANNEL1
:ACCL:ZOOM 1.000
```

:CHANnel<x>:CAN?

Function Queries all the settings of a CAN bus monitor module.

Syntax
```
:CHANnel<x>:CAN?
<x> = 13 to 16
```

:CHANnel<x>:CAN:OSOut?

Function Queries all the one-shot output settings of a CAN bus monitor module.

Syntax
```
:CHANnel<x>:CAN:OSOut?
<x> = 13 to 16
```

:CHANnel<x>:CAN:OSOut:EXECute (One Shot Out)

Function Executes one-shot output on a CAN bus monitor module.

Syntax
```
:CHANnel<x>:CAN:OSOut:EXECute
<x> = 13 to 16
```

Example
```
:CHANNEL13:CAN:OSOUT:EXECUTE
```

:CHANnel<x>:CAN:OSOut:DATA<x>

Function Sets or queries the data frame value used by a CAN bus monitor module when it performs one-shot output.

Syntax
```
:CHANnel<x>:CAN:OSOut:
DATA<x> {<String>}
CHANnel<x>'s <x> = 13 to 16
DATA<x>'s <x> = 1 to 8
<String> = "0" to "FF"
```

Example
```
:CHANNEL13:CAN:OSOUT:DATA1 FF
:CHANNEL13:CAN:OSOUT:DATA1? -> :
CHANNEL13:CAN:OSOUT:DATA1 FF
```

Description Error 151, "Invalid string data," will occur if an invalid string is specified.

:CHANnel<x>:CAN:OSOut:DLC (Data Length Code)

Function Sets or queries the byte size of the data section of the data frame used for one-shot output.

Syntax
```
:CHANnel<x>:CAN:OSOut:DLC {<NRf>}
<x> = 13 to 16
<NRf> = 0 to 15
```

Example
```
:CHANNEL13:CAN:OSOUT:DLC 15
:CHANNEL13:CAN:OSOUT:DLC? -> :
CHANNEL13:CAN:OSOUT:DLC 15
```

:CHANnel<x>:CAN:OSOut:FRAMe

Function Sets or queries the format (data or remote) of the frame used for one-shot output.

Syntax
```
:CHANnel<x>:CAN:OSOut:
FRAMe {DATA|REMote}
<x> = 13 to 16
```

Example
```
:CHANNEL13:CAN:OSOUT:FRAME DATA
:CHANNEL13:CAN:OSOUT:FRAME? -> :
CHANNEL13:CAN:OSOUT:FRAME DATA
```

:CHANnel<x>:CAN:OSOut:MFORmat (Message Format)

Function Sets or queries the message format (standard or extended) of the frame used for one-shot output.

Syntax
```
:CHANnel<x>:CAN:OSOut:
MFORmat {EXTended|STANdard}
<x> = 13 to 16
```

Example
```
:CHANNEL13:CAN:OSOUT:
MFORMAT EXTENDED
:CHANNEL13:CAN:OSOUT:MFORMAT? -> :
CHANNEL13:CAN:OSOUT:MFORMAT EXTENDED
```

:CHANnel<x>:CAN:OSOut:MID (Message ID)

Function Sets or queries the message ID of the frame used for one-shot output.

Syntax
```
:CHANnel<x>:CAN:OSOut:MID {<String>}
<x> = 13 to 16
```
- When the :CHANnel<x>:CAN:OSOut:MFORmat command is set to STANdard
  <String> = "0" to "7FF"
- When the :CHANnel<x>:CAN:OSOut:MFORmat command is set to EXTended
  <String> = "0" to "1FFFFFFF"

Example
```
:CHANNEL13:CAN:OSOUT:MID 7FF
:CHANNEL13:CAN:OSOUT:MID? -> :
CHANNEL13:CAN:OSOUT:MID 7FF
```

:CHANnel<x>:CAN:PORT?

Function Queries all the port settings of a CAN bus monitor module.

Syntax
```
:CHANnel<x>:CAN:PORT?
<x> = 13 to 16
```

:CHANnel<x>:CAN:PORT:BRATe

Function Sets or queries the bitrate of a CAN bus signal port.

Syntax
```
:CHANnel<x>:CAN:PORT:BRATe {<NRf>}
<x> = 13 to 16
<NRf> = 10000, 20000, 33300, 50000,
62500, 66700, 83300, 125000, 250000,
500000, 800000, 1000000
```

Example
```
:CHANNEL13:CAN:PORT:BRATE 33300
:CHANNEL13:CAN:PORT:BRATE? -> :
CHANNEL13:CAN:PORT:BRATE 33300
```

:CHANnel<x>:CAN:PORT:BSNum (Bit sample num)

Function Sets or queries the number of sample points for each bit of a CAN bus signal port.

Syntax
```
:CHANnel<x>:CAN:PORT:BSNum {<NRf>}
<x> = 13 to 16
<NRf> = 1 to 3
```

Example
```
:CHANNEL13:CAN:PORT:BSNUM 1
:CHANNEL13:CAN:PORT:BSNUM? -> :
CHANNEL13:CAN:PORT:BSNUM 1
```

:CHANnel<x>:CAN:PORT:LONLy

Function Sets or queries the Listen Only setting for a CAN bus monitor module port.

Syntax
```
:CHANnel<x>:CAN:PORT:LONLy {Boolean}
<x> = 13 to 16
```

Example
```
:CHANNEL13:CAN:PORT:LONLY ON
:CHANNEL13:CAN:PORT:LONLY? -> :
CHANNEL13:CAN:PORT:LONLY 1
```

:CHANnel<x>:CAN:PORT:SJW (Sync jump width)

Function Sets or queries the resynchronization jump width (also referred to as the sync jump width).

Syntax
```
:CHANnel<x>:CAN:PORT:SJW {<NRf>}
<x> = 13 to 16
<NRf> = 1 to 4
```

Example
```
:CHANNEL13:CAN:PORT:SJW 1
:CHANNEL13:CAN:PORT:SJW? -> :
CHANNEL13:CAN:PORT:SJW 1
```

Description When the sample point is set to 85%, the DL850/DL850V will not operate if <NRf> is set to 4. In this case, the DL850/DL850V will operate if <NRf> is set to 3. (Even if you change the sample point setting, the SJW setting does not change.)

:CHANnel<x>:CAN:PORT:SPOint

Function Sets or queries the sample point for each bit of a CAN bus signal port.

Syntax
```
:CHANnel<x>:CAN:PORT:SPOint {<NRf>}
<x> = 13 to 16
<NRf> = 71, 78, or 85
```

Example
```
:CHANNEL13:CAN:PORT:SPOINT 71
:CHANNEL13:CAN:PORT:SPOINT? -> :
CHANNEL13:CAN:PORT:SPOINT 71
```

Description This command is valid when the :CHANnel<x>:CAN:PORT:BRATe command is set to MANual.

:CHANnel<x>:CAN:PORT:TERMinator

Function Sets or queries whether the 124 Ω terminator between CAN_H and CAN_L on the CAN bus line is ON or OFF.

Syntax
```
:CHANnel<x>:CAN:PORT:TERMinator
{Boolean}
<x> = 13 to 16
```

Example
```
:CHANNEL13:CAN:PORT:TERMINATOR ON
:CHANNEL13:CAN:PORT:TERMINATOR? -> :
CHANNEL13:CAN:PORT:TERMINATOR 1
```

:CHANnel<x>:CAN:SCHannel<x>?

Function Queries all the settings of a sub channel on a CAN bus monitor module.

Syntax
```
:CHANnel<x>:CAN:SCHannel<x>?
CHANnel<x>'s <x> = 13 to 16
SCHANnel<x>'s <x> = 1 to 16
```

:CHANnel<x>:CAN:SCHannel<x>:INPut

Function Sets or queries whether a sub channel on a CAN bus monitor module is ON or OFF.

Syntax
```
:CHANnel<x>:CAN:SCHannel<x>:INPut
{<Boolean>}
CHANnel<x>'s <x> = 13 to 16
SCHANnel<x>'s <x> = 1 to 16
```

Example
```
:CHANNEL13:CAN:SCHANNEL1:INPUT ON
:CHANNEL13:CAN:SCHANNEL1:INPUT? -> :
CHANNEL13:CAN:SCHANNEL1:INPUT 1
```

:CHANnel<x>:CAN:SCHannel<x>:MFORmat

Function Sets or queries the message format of a sub channel on a CAN bus monitor module.

Syntax
```
:CHANnel<x>:CAN:SCHannel<x>:MFORmat
{EXTended|STANdard}
CHANnel<x>'s <x> = 13 to 16
SCHANnel<x>'s <x> = 1 to 16
```

Example
```
:CHANNEL13:CAN:SCHANNEL1:
MFORMAT STANDARD
:CHANNEL13:CAN:SCHANNEL1:MFORMAT? ->
:CHANNEL13:CAN:SCHANNEL1:
MFORMAT STANDARD
```

:CHANnel<x>:CAN:SCHannel<x>:MID

Function Sets or queries the message ID of a sub channel on a CAN bus monitor module.

Syntax
```
:CHANnel<x>:CAN:SCHannel<x>:
MID {<String>}
CHANnel<x>'s <x> = 13 to 16
SCHANnel<x>'s <x> = 1 to 16
```
- When the :CHANnel<x>:CAN:OSOut:MFORmat command is set to STANdard
  <String> = "0" to "7FF"
- When the :CHANnel<x>:CAN:OSOut:MFORmat command is set to EXTended
  <String> = "0" to "1FFFFFFF"

Example
```
:CHANNEL13:CAN:SCHANNEL1:MID 7FF
:CHANNEL13:CAN:SCHANNEL1:MID? -> :
CHANNEL1:CAN:SCHANNEL1:MID 7FF
```

Description Error 151, "Invalid string data," will occur if an invalid string is specified.

:CHANnel<x>:CAN:SCHannel<x>:SBIT (Start BIT)

Function Sets or queries the bit number of the starting position of a sub channel on a CAN bus monitor module.

Syntax
```
:CHANnel<x>:CAN:SCHannel<x>:
SBIT {<NRf>}
CHANnel<x>'s <x> = 13 to 16
SCHANnel<x>'s <x> = 1 to 16
<NRf> = 0 to 63
```

Example
```
:CHANNEL13:CAN:SCHANNEL1:SBIT 10
:CHANNEL13:CAN:SCHANNEL1:SBIT? -> :
CHANNEL13:CAN:SCHANNEL1:SBIT 10
```

:CHANnel<x>:CAN:SCHannel<x>:BICount (Bit Count)

Function Sets or queries the bit length of a sub channel on a CAN bus monitor module.

Syntax
```
:CHANnel<x>:CAN:SCHannel<x>:
BICount {<NRf>}
CHANnel<x>'s <x> = 13 to 16
SCHANnel<x>'s <x> = 1 to 16
<NRf> = 2 to 16 (2 to 32 on firmware
versions 1.2 and later)
```

Example
```
:CHANNEL13:CAN:SCHANNEL1:BICOUNT 10
:CHANNEL13:CAN:SCHANNEL1:BICOUNT? ->
:CHANNEL13:CAN:SCHANNEL1:BICOUNT 10
```

:CHANnel<x>:CAN:SCHannel<x>:BORDer (Byte Order)

Function Sets or queries the endianness used to save to internal memory the data of a sub channel on a CAN bus module.

Syntax
```
:CHANnel<x>:CAN:SCHannel<x>:
BORDer {BIG|LITTle}
CHANnel<x>'s <x> = 13 to 16
SCHANnel<x>'s <x> = 1 to 16
```

Example
```
:CHANNEL13:CAN:SCHANNEL1:BORDER BIG
:CHANNEL13:CAN:SCHANNEL1:BORDER? ->
:CHANNEL13:CAN:SCHANNEL1:BORDER BIG
```

:CHANnel<x>:CAN:SCHannel<x>:VTYPe (Value Type)

Function Sets or queries the data type of a sub channel on a CAN bus monitor module.

Syntax
```
:CHANnel<x>:CAN:SCHannel<x>:
VTYPe {UNSigned|SIGNed|LOGic}
({UNSigned|FLOat|SIGNed|LOGic} on
firmware versions 1.2 and later)
CHANnel<x>'s <x> = 13 to 16
SCHANnel<x>'s <x> = 1 to 16
```

Example
```
:CHANNEL13:CAN:SCHANNEL1:
VTYPE SIGNED
:CHANNEL13:CAN:SCHANNEL1:VTYPE? -> :
CHANNEL13:CAN:SCHANNEL1:VTYPE SIGNED
```

:CHANnel<x>:CAN:SCHannel<x>:BCOunt (Byte Count)

Function Sets or queries the byte count of a sub channel on a CAN bus monitor module.

Syntax
```
:CHANnel<x>:CAN:SCHannel<x>:
BCOunt {AUTO|<NRf>}
CHANnel<x>'s <x> = 13 to 16
SCHANnel<x>'s <x> = 1 to 16
<NRf> = 1 to 8
```

Example
```
:CHANNEL13:CAN:SCHANNEL1:BCOUNT 8
:CHANNEL13:CAN:SCHANNEL1:BCOUNT? ->
:CHANNEL13:CAN:SCHANNEL1:BCOUNT 8
```

:CHANnel<x>:CAN:SCHannel<x>:FACTor

Function Sets or queries the scaling constant (value per pit) of a sub channel on a CAN bus monitor module.

Syntax
```
:CHANnel<x>:CAN:SCHannel<x>:
FACTor {<NRf>}
CHANnel<x>'s <x> = 13 to 16
SCHANnel<x>'s <x> = 1 to 16
<NRf> = -9.9999E+30 to 9.9999E+30
```

Example
```
:CHANNEL13:CAN:SCHANNEL1:FACTOR 5E15
:CHANNEL13:CAN:SCHANNEL1:FACTOR? ->
:CHANNEL13:CAN:SCHANNEL1:
FACTOR 5.00000E+15
```

Description This command is valid when the data type (the :CHANnel<x>:CAN:SCHannel<x>:VTYPe command) is set to UNSigned or SIGNed.

:CHANnel<x>:CAN:SCHannel<x>:OFFSet

Function Sets or queries the scaling constant (offset value) of a sub channel on a CAN bus monitor module.

Syntax
```
:CHANnel<x>:CAN:SCHannel<x>:
OFFSet {<NRf>}
CHANnel<x>'s <x> = 13 to 16
SCHANnel<x>'s <x> = 1 to 16
<NRf> = -9.9999E+30 to 9.9999E+30
```

Example
```
:CHANNEL13:CAN:SCHANNEL1:OFFSET 5E15
:CHANNEL13:CAN:SCHANNEL1:OFFSET? ->
:CHANNEL13:CAN:SCHANNEL1:
OFFSET 5.00000E+15
```

Description This command is valid when the data type (the :CHANnel<x>:CAN:SCHannel<x>:VTYPe command) is set to UNSigned or SIGNed.

:CHANnel<x>:CAN:SCHannel<x>:SCALe

Function Sets or queries the display range values (upper and lower limits) of a sub channel on a CAN bus monitor module.

Syntax
```
:CHANnel<x>:CAN:SCHannel<x>:
SCALe {AUTO|DEFault|<NRf>,<NRf>}
CHANnel<x>'s <x> = 13 to 16
SCHANnel<x>'s <x> = 1 to 16
<NRf> = -30.000E+30 to 30.000E+30
```

Example
```
:CHANNEL13:CAN:SCHANNEL1:SCALE AUTO
:CHANNEL13:CAN:SCHANNEL1:SCALE? -> :
CHANNEL13:CAN:SCHANNEL1:SCALE AUTO
```

Description This command is valid when the data type (the :CHANnel<x>:CAN:SCHannel<x>:VTYPe command) is set to UNSigned, SIGNed, or FLOat.

:CHANnel<x>:CAN:SCHannel<x>:POSition

Function Sets or queries the vertical position of a sub channel on a CAN bus monitor module.

Syntax
```
:CHANnel<x>:CAN:SCHannel<x>:
POSition {<NRf>}
CHANnel<x>'s <x> = 13 to 16
SCHANnel<x>'s <x> = 1 to 16
<NRf> = -5.00 to 5.00 (div; in 0.01
steps)
```

Example
```
:CHANNEL13:CAN:SCHANNEL1:POSITION 1
:CHANNEL13:CAN:SCHANNEL1:POSITION?
-> :CHANNEL13:CAN:SCHANNEL1:
POSITION 1.00
```

:CHANnel<x>:CAN:SCHannel<x>:UNIT

Function Sets or queries the unit of data of a sub channel on a CAN bus monitor module.

Syntax
```
:CHANnel<x>:CAN:SCHannel<x>:
UNIT {<String>}
CHANnel<x>'s <x> = 13 to 16
SCHANnel<x>'s <x> = 1 to 16
<String> = Up to 8 characters
```

Example
```
:CHANNEL13:CAN:SCHANNEL1:UNIT "AAAA"
:CHANNEL13:CAN:SCHANNEL1:UNIT? -> :
CHANNEL13:CAN:SCHANNEL1:UNIT "AAAA"
```

Description This command is valid when the data type (the :CHANnel<x>:CAN:SCHannel<x>:VTYPe command) is set to UNSigned, SIGNed, or FLOat.

:CHANnel<x>:CAN:SCHannel<x>:BLABel<x>

Function Sets or queries a bit label of a sub channel on a CAN bus monitor module when the sub channel's data type is logic.

Syntax
```
:CHANnel<x>:CAN:SCHannel<x>:
BLABel<x> {<String>}
CHANnel<x>'s <x> = 13 to 16
SCHANnel<x>'s <x> = 1 to 16
BLABel<x>'s <x> = 1 to 8
<String> = Up to 16 characters
```

Example
```
:CHANNEL13:CAN:SCHANNEL1:
BLABEL1 "AAA"
:CHANNEL13:CAN:SCHANNEL1:BLABEL1? ->
:CHANNEL13:CAN:SCHANNEL1:
BLABEL1 "AAA"
```

Description This command is valid when the data type (the :CHANnel<x>:CAN:SCHannel<x>:VTYPe command) is set to LOGic.

:CHANnel<x>:CAN:SCHannel<x>:LABel

Function Sets or queries the label of a sub channel on a CAN bus monitor module.

Syntax
```
:CHANnel<x>:CAN:SCHannel<x>:
LABel {<String>}
:CHANnel<x>:CAN:SCHannel<x>:LABel?
CHANnel<x>'s <x> = 13 to 16
SCHANnel<x>'s <x> = 1 to 16 (1 to 60
on firmware versions 1.2 and later)
<String> = Up to 16 characters
```

Example
```
:CHANNEL13:CAN:SCHANNEL1:
LABEL "AAAA"
:CHANNEL13:CAN:SCHANNEL1:LABEL? -> :
CHANNEL13:CAN:SCHANNEL1:LABEL "AAAA"
```

:CHANnel<x>:DISPlay

Function Sets or queries whether the channel is displayed.

Syntax
```
:CHANnel<x>:DISPlay {<Boolean>}
:CHANnel<x>:DISPlay?
<x> = 1 to 16
```

Example
```
:CHANNEL1:DISPLAY 1
:CHANNEL1:DISPLAY? -> :CHANNEL1
:DISPLAY 1
```

Description Error 241, “Hardware missing,” will occur if there is no module installed in the slot that corresponds to the specified channel.

:CHANnel<x>:FREQ?

Function Queries all the settings of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ?
<x> = 1 to 16
```

:CHANnel<x>:FREQ:INPut?

Function Queries all the input settings of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:INPut?
<x> = 1 to 16
```

:CHANnel<x>:FREQ:INPut:BWIDth

Function Sets or queries the input signal bandwidth limit of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:INPut:BWIDth {FULL|
<Frequency>
:CHANnel<x>:FREQ:INPut:BWIDth?
<x> = 1 to 16
<Frequency> = 100Hz, 1kHz, 10kHz,
100kHz
If Preset is set to AC200V or
AC100V, you cannot select FULL.
```

Example
```
:CHANNEL1:FREQ:INPUT:BWIDTH FULL
:CHANNEL1:FREQ:INPUT:BWIDTH? ->
:CHANNEL1:FREQ:INPUT:BWIDTH FULL
```

:CHANnel<x>:FREQ:INPut:CELimination

Function Sets or queries the input signal chattering elimination of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:INPut:
CELimination {<Time>}
:CHANnel<x>:FREQ:INPut:CELimination?
<x> = 1 to 16
<Time> = 0 to 1000ms
```

Example
```
:CHANNEL1:FREQ:INPUT:
CELIMINATION 0.000
:CHANNEL1:FREQ:INPUT:
CELIMINATION? -> :CHANNEL1:FREQ:
INPUT:CELIMINATION 0.000
```

:CHANnel<x>:FREQ:INPut:COUPling

Function Sets or queries the input coupling setting of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:INPut:COUPling {AC|DC}
:CHANnel<x>:FREQ:INPut:COUPling?
<x> = 1 to 16
```

Example
```
:CHANNEL1:FREQ:INPUT:COUPLING DC
:CHANNEL1:FREQ:INPUT:COUPLING? ->
:CHANNEL1:FREQ:INPUT:COUPLING DC
```

:CHANnel<x>:FREQ:INPut:HYSTeresis

Function Sets or queries the hysteresis of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:INPut:
HYSTeresis {HIGH|LOW|MIDDle}
:CHANnel<x>:FREQ:INPut:HYSTeresis?
<x> = 1 to 16
```

Example
```
:CHANNEL1:FREQ:INPUT:HYSTERESIS LOW
:CHANNEL1:FREQ:INPUT:HYSTERESIS? ->
:CHANNEL1:FREQ:INPUT:HYSTERESIS LOW
```

:CHANnel<x>:FREQ:INPut:PRESet

Function Sets or queries the preset setting of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:INPut:
PRESet {AC100v|AC200v|EMPickup|LOG12v|
LOG24v|LOG3v|LOG5v|PULLup|USER|ZERO}
:CHANnel<x>:FREQ:INPut:PRESet?
<x> = 1 to 16
```

Example
```
:CHANNEL1:FREQ:INPUT:PRESET USER
:CHANNEL1:FREQ:INPUT:PRESET? ->
:CHANNEL1:FREQ:INPUT:PRESET USER
```

:CHANnel<x>:FREQ:INPut:PROBe

Function Sets or queries the probe attenuation of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:INPut:PROBe {<NRf>}
:CHANnel<x>:FREQ:INPut:PROBe?
<x> = 1 to 16
<NRf> = 1, 10
```

Example
```
:CHANNEL1:FREQ:INPUT:PROBE 1
:CHANNEL1:FREQ:INPUT:PROBE? ->
:CHANNEL1:FREQ:INPUT:PROBE 1
```

:CHANnel<x>:FREQ:INPut:PULLup

Function Sets or queries whether pull-up is turned on for an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:INPut:
PULLup {<Boolean>}
:CHANnel<x>:FREQ:INPut:PULLup?
<x> = 1 to 16
```

Example
```
:CHANNEL1:FREQ:INPUT:PULLUP ON
:CHANNEL1:FREQ:INPUT:PULLUP? ->
:CHANNEL1:FREQ:INPUT:PULLUP 1
```

:CHANnel<x>:FREQ:INPut:SLOPe

Function Sets or queries the input slope of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:INPut:SLOPe {FALL|RISE}
:CHANnel<x>:FREQ:INPut:SLOPe?
<x> = 1 to 16
```

Example
```
:CHANNEL1:FREQ:INPUT:SLOPE RISE
:CHANNEL1:FREQ:INPUT:SLOPE? ->
:CHANNEL1:FREQ:INPUT:SLOPE RISE
```

:CHANnel<x>:FREQ:INPut:THReshold

Function Sets or queries the threshold level of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:INPut:
THReshold {<Voltage>}
:CHANnel<x>:FREQ:INPut:THReshold?
<x> = 1 to 16
<Voltage> = 100 to -100V
```

Example
```
:CHANNEL1:FREQ:INPUT:THRESHOLD
0.00000E+00
:CHANNEL1:FREQ:INPUT:THRESHOLD? ->
:CHANNEL1:FREQ:INPUT:
THRESHOLD 0.00000E+00
```

:CHANnel<x>:FREQ:INPut:VRANge

Function Sets or queries the voltage range of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:INPut:
VRANge {<Voltage>}
:CHANnel<x>:FREQ:INPut:VRANge?
<x> = 1 to 16
<Voltage> = 1 to 500V
```

Example
```
:CHANNEL1:FREQ:INPUT:VRANGE 10
:CHANNEL1:FREQ:INPUT:VRANGE? ->
:CHANNEL1:FREQ:INPUT:VRANGE 10
```

:CHANnel<x>:FREQ:LSCale?

Function Queries all the linear scaling settings of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:LSCale?
```

:CHANnel<x>:FREQ:LSCale:AVALue

Function Sets or queries the scaling coefficient A that is used during linear scaling for an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:LSCale:
AVALue {<NRf>}
:CHANnel<x>:FREQ:LSCale:AVALue?
<x> = 1 to 16
<NRf> = -9.9999E+30 to -1E-30, 1E-30
to 9.9999E+30
```

Example
```
:CHANNEL1:FREQ:LSCALE:
AVALUE 25.0000E+00
:CHANNEL1:FREQ:LSCALE:AVALUE? -> :
CHANNEL1:FREQ:LSCALE:
AVALUE 25.0000E+00
```

Description The coefficient cannot be set to 0.

:CHANnel<x>:FREQ:LSCale:BVALue

Function Sets or queries the scaling coefficient B that is used during linear scaling for an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:LSCale:
BVALue {<NRf>}
:CHANnel<x>:FREQ:LSCale:BVALue?
<x> = 1 to 16
<NRf> = -9.9999E+30 to -1E-30, 1E-30
to 9.9999E+30
```

Example
```
:CHANNEL1:FREQ:LSCALE:
BVALUE -25.0000E+00
:CHANNEL1:FREQ:LSCALE:BVALUE? ->
:CHANNEL1:FREQ:LSCALE:
BVALUE -25.0000E+00
```

:CHANnel<x>:FREQ:LSCale:GETMeasure

Function Executes the measurement of the linear scaling P1X or P2X value of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:LSCale:
GETMeasure {P1X|P2X}
<x> = 1 to 16
```

Example
```
:CHANnel1:FREQ:LSCale:GETMeasure P1X
```

:CHANnel<x>:FREQ:LSCale:MODE

Function Sets or queries the linear scale mode of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:LSCale:MODE {AXB|
OFF|P12}
:CHANnel<x>:FREQ:LSCale:MODE?
<x> = 1 to 16
```

Example
```
:CHANNEL1:FREQ:LSCALE:MODE AXB
:CHANNEL1:FREQ:LSCALE:MODE? ->
:CHANNEL1:FREQ:LSCALE:MODE AXB
```

:CHANnel<x>:FREQ:LSCale:{P1X|P1Y|P2X|P2Y}

Function Sets or queries the linear scaling P1X, P1Y, P2X, or P2Y value of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:LSCale:{P1X|P1Y|
P2X|P2Y} {<NRf>}
:CHANnel<x>:FREQ:LSCale:{P1X|P1Y|
P2X|P2Y}?
<x> = 1 to 16
P1X or P2X's <NRf> = -9.9999E+30 to
-1E-30, 0, 1E-30 to 9.9999E+30
P1Y or P2Y's <NRf> = -9.9999E+25 to
-1E-25, 0, 1E-25 to 9.9999E+25
```

Example
```
:CHANNEL1:FREQ:LSCALE:
P1X 1.00000E+00
:CHANNEL1:FREQ:LSCALE:P1X? ->
:CHANNEL1:FREQ:LSCALE:
P1X 1.00000E+00
```

:CHANnel<x>:FREQ:LSCale:UNIT

Function Sets or queries the linear-scaling unit string of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:LSCale:
UNIT {<String>}
:CHANnel<x>:FREQ:LSCale:UNIT?
<x> = 1 to 16
<String> = Up to 4 characters
```

Example
```
:CHANNEL1:FREQ:LSCALE:UNIT "UU"
:CHANNEL1:FREQ:LSCALE:UNIT? ->
:CHANNEL1:FREQ:LSCALE:UNIT "UU"
```

:CHANnel<x>:FREQ:OFFSet

Function Sets or queries the offset of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:OFFSet {<NRf>|
<Frequency>|<Time>}
:CHANnel<x>:FREQ:OFFSet?
<x> = 1 to 16
```

Example
```
:CHANNEL5:FREQ:OFFSET 1
:CHANNEL5:FREQ:OFFSET? ->
:CHANNEL5:FREQ:OFFSET 0.000000E+00
```

Description `{<NRf>|<Frequency>|<Time>}` = The selectable range varies depending on the range setting. See the DL850/DL850V Features Guide for this information.

:CHANnel<x>:FREQ:POSition

Function Sets or queries the vertical position for an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:POSition {<NRf>}
:CHANnel<x>:FREQ:POSition?
<x> = 1 to 16
<NRf> = -5.00 to 5.00 (div; in 0.01
div steps)
```

Example
```
:CHANNEL5:FREQ:POSITION 1.00
:CHANNEL5:FREQ:POSITION? ->
:CHANNEL5:FREQ:POSITION 0.00
```

:CHANnel<x>:FREQ:SCALe

Function Sets or queries the upper and lower limits of the screen for an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:
SCALe {<NRf>,<NRf>|
<Frequency>,<Frequency>|
<Time>,<Time>}
:CHANnel<x>:FREQ:SCALe?
<x> = 1 to 16
<NRf>, <Frequency>, <Time> = The
selectable range varies depending on
the range setting.
```

Example
```
:CHANNEL5:FREQ:SCALE 5000,-5000
:CHANNEL5:FREQ:SCALE? ->
:CHANNEL5:FREQ:SCALE 5.000000E+03,
-5.000000E+03
```

:CHANnel<x>:FREQ:SETup?

Function Queries all the FV settings of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:SETup?
<x> = 1 to 16
```

:CHANnel<x>:FREQ:SETup:CFRequency

Function Sets or queries the center frequency of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:SETup:
CFRequency {<Frequency>}
:CHANnel<x>:FREQ:SETup:CFRequency?
<x> = 1 to 16
<Frequency> = 50Hz, 60Hz, 400Hz
```

Example
```
:CHANnel5:FREQ:SETup:CFRequency 60
:CHANnel5:FREQ:SETup:CFRequency? ->
:CHANnel<x>:FREQ:SETup:CFRequency 60
```

:CHANnel<x>:FREQ:SETup:DECeleration

Function Sets or queries whether deceleration prediction is turned on for an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:SETup:
DECeleration {<Boolean>}
:CHANnel<x>:FREQ:SETup:DECeleration?
<x> = 1 to 16
```

Example
```
:CHANNEL5:FREQ:SETUP:DECELERATION ON
:CHANNEL5:FREQ:SETUP:DECELERATION?
-> :CHANNEL5:FREQ:SETUP:
DECELERATION 1
```

:CHANnel<x>:FREQ:SETup:DPULse

Function Sets or queries the distance per pulse of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:SETup:
DPULse {<NRf>}
:CHANnel<x>:FREQ:SETup:DPULse?
<x> = 1 to 16
<NRf> = 9.9999E+30 to -9.9999E+30
```

Example
```
:CHANNEL5:FREQ:SETUP:
DPULSE 1.00000E+00
:CHANNEL5:FREQ:SETUP:DPULSE? ->
:CHANNEL5:FREQ:SETUP:
DPULSE 1.00000E+00
```

:CHANnel<x>:FREQ:SETup:FILTer?

Function Queries all the filter settings of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:SETup:FILTer?
```

:CHANnel<x>:FREQ:SETup:FILTer:SMOothing?

Function Queries all the smoothing settings of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:SETup:FILTer:
SMOothing?
```

:CHANnel<x>:FREQ:SETup:FILTer:SMOothing:MODE

Function Sets or queries whether smoothing is turned on for an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:SETup:FILTer:
SMOothing:MODE {<Boolean>}
:CHANnel<x>:FREQ:SETup:FILTer:
SMOothing:MODE?
<x> = 1 to 16
```

Example
```
:CHANNEL5:FREQ:SETUP:FILTER:
SMOOTHING:MODE ON
:CHANNEL5:FREQ:SETUP:FILTER:
SMOOTHING:MODE? -> :CHANNEL5:FREQ:
SETUP:FILTER:SMOOTHING:MODE 1
```

:CHANnel<x>:FREQ:SETup:FILTer:SMOothing:VALue

Function Sets or queries the moving average order of smoothing of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:SETup:FILTer:
SMOothing:VALue {<Time>}
:CHANnel<x>:FREQ:SETup:FILTer:
SMOothing:VALue?
<x> = 1 to 16
<Time> = 0ms to 1000.0ms
```

Example
```
:CHANNEL5:FREQ:SETUP:FILTER:
SMOOTHING:VALUE 0.1
:CHANNEL5:FREQ:SETUP:FILTER:
SMOOTHING:VALUE? -> :CHANNEL5:FREQ:
SETUP:FILTER:SMOOTHING:VALUE 0.1000
```

:CHANnel<x>:FREQ:SETup:FILTer:PAVerage?

Function Queries all the pulse averaging settings of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:SETup:FILTer:
PAVerage?
```

:CHANnel<x>:FREQ:SETup:FILTer:PAVerage:MODE

Function Sets or queries whether pulse averaging is turned on for an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:SETup:FILTer:
PAVerage:MODE {<Boolean>}
:CHANnel<x>:FREQ:SETup:FILTer:
PAVerage:MODE?
<x> = 1 to 16
```

Example
```
:CHANNEL5:FREQ:SETUP:FILTER:
PAVERAGE:MODE ON
:CHANNEL5:FREQ:SETUP:FILTER:
PAVERAGE:MODE? -> :CHANNEL5:FREQ:
SETUP:FILTER:PAVERAGE:MODE 1
```

:CHANnel<x>:FREQ:SETup:FILTer:PAVerage:VALue

Function Sets or queries the number of pulses to average over for an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:SETup:FILTer:
PAVerage:VALue {<NRf>}
:CHANnel<x>:FREQ:SETup:FILTer:
PAVerage:VALue?
<x> = 1 to 16
<NRf> = 1 to 4096
```

Example
```
:CHANNEL5:FREQ:SETUP:FILTER:
PAVERAGE:VALUE 2
:CHANNEL5:FREQ:SETUP:FILTER:
PAVERAGE:VALUE? -> :CHANNEL5:FREQ:
SETUP:FILTER:PAVERAGE:VALUE 2
```

:CHANnel<x>:FREQ:SETup:FUNCtion

Function Sets or queries the measurement mode of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:SETup:
FUNCtion {DUTY|FREQuency|PERiod|
PFReq|PINTeg|PWIDth|RPM|RPS|
VELocity}
:CHANnel<x>:FREQ:SETup:FUNCtion?
<x> = 1 to 16
```

Example
```
:CHANNEL5:FREQ:SETUP:FUNCTION
FREQUENCY
:CHANNEL5:FREQ:SETUP:FUNCTION
FREQUENCY? -> :CHANNEL5:FREQ:SETUP:
FUNCTION FREQUENCY
```

:CHANnel<x>:FREQ:SETup:LRESet

Function Sets or queries whether over-limit reset is turned on for an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:SETup:
LRESet {<Boolean>}
:CHANnel<x>:FREQ:SETup:LRESet?
```

Example
```
:CHANNEL5:FREQ:SETUP:LRESET ON
:CHANNEL5:FREQ:SETUP:LRESET? ->
:CHANNEL5:FREQ:SETUP:LRESET 1
```

:CHANnel<x>:FREQ:SETup:MPULse

Function Sets or queries whether the measurement pulse is positive or negative for an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:SETup:
MPULse {POSitive|NEGative}
:CHANnel<x>:FREQ:SETup:MPULse?
```

Example
```
:CHANNEL5:FREQ:SETUP:MPULSE POSITIVE
:CHANNEL5:FREQ:SETUP:MPULSE? ->
:CHANNEL5:FREQ:SETUP:MPULSE POSITIVE
```

:CHANnel<x>:FREQ:SETup:PROTate

Function Sets or queries the number of pulses per rotation of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:SETup:
PROTate {<NRf>}
:CHANnel<x>:FREQ:SETup:PROTate?
<NRf> = 1 to 99999
```

Example
```
:CHANNEL5:FREQ:SETUP:PROTATE 100
:CHANNEL5:FREQ:SETUP:PROTATE? ->
:CHANNEL5:FREQ:SETUP:PROTATE 100
```

:CHANnel<x>:FREQ:SETup:RESet

Function Resets the pulse count of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:SETup:RESet
```

Example
```
:CHANNEL5:FREQ:SETUP:RESET
```

:CHANnel<x>:FREQ:SETup:STOPpredict

Function Sets or queries whether stop prediction is turned on for an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:SETup:
STOPpredict {<NRf>|OFF}
:CHANnel<x>:FREQ:SETup:STOPpredict?
<NRf> = 1.5, 2, 3, 4, 5, 6, 7, 8, 9,
10
```

Example
```
:CHANNEL5:FREQ:SETUP:STOPPREDICT 10
:CHANNEL5:FREQ:SETUP:STOPPREDICT? ->
:CHANNEL5:FREQ:SETUP:STOPPREDICT 10
```

:CHANnel<x>:FREQ:SETup:TUNit

Function Sets or queries the time unit (when measuring velocity) of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:SETup:TUNit {HOUR|
MIN|SEC}
:CHANnel<x>:FREQ:SETup:TUNit?
```

Example
```
:CHANNEL5:FREQ:SETUP:TUNIT MIN
:CHANNEL5:FREQ:SETUP:TUNIT? ->
:CHANNEL5:FREQ:SETUP:TUNIT MIN
```

:CHANnel<x>:FREQ:SETup:UNIT

Function Sets or queries the pulse integration unit of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:SETup:
UNIT {<String>}
:CHANnel<x>:FREQ:SETup:UNIT?
<String> = Up to 4 characters
```

Example
```
:CHANNEL5:FREQ:SETUP:UNIT "ABC"
:CHANNEL5:FREQ:SETUP:UNIT? ->
:CHANNEL5:FREQ:SETUP:UNIT "ABC"
```

:CHANnel<x>:FREQ:SETup:UPULse

Function Sets or queries the unit/pulse setting of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:SETup:
UPULse {<NRf>}
:CHANnel<x>:FREQ:SETup:UPULse?
<NRf> = 9.9999E+30 to -9.9999E+30
```

Example
```
:CHANNEL5:FREQ:SETUP:UPULSE 3
:CHANNEL5:FREQ:SETUP:UPULSE? ->
:CHANNEL5:FREQ:SETUP:UPULSE 3
```

:CHANnel<x>:FREQ:SETup:VUNit

Function Sets or queries the velocity unit of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:SETup:
VUNit {<String>}
:CHANnel<x>:FREQ:SETup:VUNit?
<String> = Up to 4 characters
```

Example
```
:CHANNEL5:FREQ:SETUP:VUNIT "m/s"
:CHANNEL5:FREQ:SETUP:VUNIT? ->
:CHANNEL5:FREQ:SETUP:VUNIT "m/s"
```

:CHANnel<x>:FREQ:VDIV

Function Sets or queries the Value/Div setting of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:VDIV {<NRf>|<Frequency>|<Time>}
:CHANnel<x>:FREQ:VDIV?
{<NRf>|<Frequency>|<Time>} = See the DL850/DL850V Features Guide for this information.
```

Example
```
:CHANNEL5:FREQ:VDIV 20
:CHANNEL5:FREQ:VDIV? -> :CHANNEL5:FREQ:VDIV 20
```

:CHANnel<x>:FREQ:ZOOM

Function Sets or queries the vertical zoom factor of an installed frequency module.

Syntax
```
:CHANnel<x>:FREQ:ZOOM {<NRf>}
:CHANnel<x>:FREQ:ZOOM?
<x> = 1 to 16
<NRf> = 0.1, 0.111, 0.125, 0.143, 0.167, 0.2, 0.25, 0.33, 0.4, 0.5, 0.556, 0.625, 0.667, 0.714, 0.8, 0.833, 1, 1.11, 1.25, 1.33, 1.43, 1.67, 2, 2.22, 2.5, 3.33
```

Example
```
:CHANNEL5:FREQ:ZOOM 1
:CHANNEL5:FREQ:ZOOM? -> :CHANNEL5:FREQ:ZOOM 1
```

:CHANnel<x>:LABel

Function Sets or queries the waveform label of a channel.

Syntax
```
:CHANnel<x>:LABel {<String>}
:CHANnel<x>:LABel?
<x> = 1 to 16
<String> = Up to 16 characters
```

Example
```
:CHANNEL5:LABEL "ABC"
:CHANNEL5:LABEL? -> :CHANNEL5:LABEL "ABC"
```

Description This setting is invalid for a module that has sub channels, but it is valid for logic modules.

:CHANnel<x>:LOGic?

Function Queries all settings of an installed logic input module.

Syntax
```
:CHANnel<x>:LOGic?
```

:CHANnel<x>:LOGic:BMAPping

Function Sets or queries the bit display method of the specified logic channel.

Syntax
```
:CHANnel<x>:LOGic:BMAPping {AUTO|FIXed}
:CHANnel<x>:LOGic:BMAPping?
<x> = 1 to 16
```

Example
```
:CHANNEL15:LOGIC:BMAPPING AUTO
:CHANNEL15:LOGIC:BMAPPING? -> :CHANNEL15:LOGIC:BMAPPING AUTO
```

:CHANnel<x>:LOGic: {BIT1|...|BIT8}?

Function Queries all settings of each bit of an installed logic input module.

Syntax
```
:CHANnel<x>:LOGic: {BIT1|...|BIT8}?
```

:CHANnel<x>:LOGic: {BIT1|...|BIT8}:DISPlay

Function Sets or queries whether the display of each bit is turned on for the specified logic channel.

Syntax
```
:CHANnel<x>:LOGic: {BIT1|...|BIT8}:DISPlay {<Boolean>}
:CHANnel<x>:LOGic: {BIT1|...|BIT8}:DISPlay?
<x> = 1 to 16
```

Example
```
:CHANNEL15:LOGIC:BIT1:DISPLAY ON
:CHANNEL15:LOGIC:BIT1:DISPLAY? -> :CHANNEL15:LOGIC:BIT1:DISPLAY 1
```

:CHANnel<x>:LOGic: {BIT1|...|BIT8}:LABel

Function Sets or queries the label of each bit for the specified logic channel.

Syntax
```
:CHANnel<x>:LOGic: {BIT1|...|BIT8}:LABel<String>
:CHANnel<x>:LOGic: {BIT1|...|BIT8}:LABel?
<String> = Up to 16 characters
<x> = 1 to 16
```

Example
```
:CHANNEL15:LOGIC:BIT8:LABEL "ABC"
:CHANNEL15:LOGIC:BIT8:LABEL? -> :CHANNEL15:LOGIC:BIT8:LABEL "ABC"
```

:CHANnel<x>:LOGic: {BIT1|...|BIT8}: CELimination

Function Sets or queries the chattering elimination for the specified bit of the specified logic channel.

Syntax
```
:CHANnel<x>:LOGic: {BIT1|...|BIT8}:
CELimination {OFF|<Time>}
:CHANnel<x>:LOGic: {BIT1|...|BIT8}:
CELimination?
<x> = 1 to 16
<Time> = 5ms, 10ms, 20ms, 50ms,
100ms
```

Example
```
:CHANNEL15:LOGIC:BIT8:
CELIMINATION 0.01
:CHANNEL15:LOGIC:BIT8:CELIMINATION?
-> :CHANNEL15:LOGIC:BIT8:
CELIMINATION 1.0000000E-02
```

:CHANnel<x>:LOGic:POSition

Function Sets or queries the vertical position of the specified logic channel.

Syntax
```
:CHANnel<x>:LOGic:POSition {<NRf>}
:CHANnel<x>:LOGic:POSition?
<x> = 1 to 16
<NRf> = -5.00 to 5.00 (div; in 0.01
div steps)
```

Example
```
:CHANNEL15:LOGIC:POSITION 1
:CHANNEL15:LOGIC:POSITION? ->
:CHANNEL15:LOGIC:POSITION 1.00
```

:CHANnel<x>:LOGic:ZOOM

Function Sets or queries the vertical zoom factor of an installed module that can perform analog voltage measurements.

Syntax
```
:CHANnel<x>[:VOLTage]:ZOOM {<NRf>}
:CHANnel<x>[:VOLTage]:ZOOM?
<x> = 1 to 16
<NRf> = 0.1, 0.111, 0.125, 0.143,
0.167, 0.2, 0.25, 0.33, 0.4, 0.5,
0.556, 0.625, 0.667, 0.714, 0.8,
0.833, 1, 1.11, 1.25, 1.33, 1.43,
1.67, 2, 2.22, 2.5, 3.33, 4, 5, 6.67,
8, 10, 12.5, 16.7, 20, 25, 40, 50,
100
```

Example
```
:CHANNEL15:LOGIC:ZOOM 1
:CHANNEL15:LOGIC:ZOOM? -> :CHANNEL15
:LOGIC:ZOOM 1.000
```

:CHANnel<x>:MODUle?

Function Queries the module that is installed in the channel.

Syntax
```
:CHANnel<x>:MODUle?
<x> = 1 to 16
```

Description Values that are returned from each module

NOMODULE	No module is installed in the specified channel or an even-numbered channel in which a 16-CH Voltage Input Module is installed was specified.
M720210	720210 (100M module)
M720220	16-CH Voltage Input Module
M720230	Logic input module
M720240	CAN bus monitor module

If a 1-channel module or a 16-CH Voltage Input Module is installed and you specify an even channel for <x>, this command will return "NOMODULE."

:CHANnel<x>:STRain?

Function Queries all settings of an installed strain module.

Syntax
```
:CHANnel<x>:STRain?
<x> = 1 to 16
```

:CHANnel<x>:STRain:BALance?

Function Queries whether balancing will be performed on an installed strain module.

Syntax
```
:CHANnel<x>:STRain:BALance?
```

Description Returns the status of all installed strain modules.

:CHANnel<x1>:STRain:BALance: CHANnel<x2>

Function Sets or queries the channels that balancing will be performed on for an installed strain module.

Syntax
```
:CHANnel<x>:STRain:BALance:
CHANnel<x> {<Boolean>}
:CHANnel<x>:STRain:BALance:
CHANnel<x>?
```

Example
```
:CHANNEL11:STRAIN:BALANCE:
CHANNEL12 ON
:CHANNEL11:STRAIN:BALANCE:CHANNEL12?
-> :CHANNEL11:STRAIN:BALANCE:
CHANNEL12 1
```

Description The channels you can specify are 1 to 16 for <x1, x2>.
However, you must specify strain module channels.

:CHANnel<x>:STRain:BALance:EXECute

Function Executes strain balancing on an installed strain module.

Syntax
```
:CHANnel<x>:STRain:BALance:EXECute
```

Description This command balances channels that are switched on with the :CHANnel<x1>:STRain:BALance:CHANnel<x2> command. <x> can be any number provided that it corresponds to the number of a channel of a strain module.

:CHANnel<x>:STRain:BWIDth

Function Sets or queries the filter of an installed strain module.

Syntax
```
:CHANnel<x>:STRain:BWIDth {FULL|
<Frequency>}
:CHANnel<x>:STRain:BWIDth?
<Frequency> = 10Hz, 100Hz, 1kHz
```

Example
```
:CHANNEL11:STRAIN:BWIDTH 10.00E+00
:CHANNEL11:STRAIN:BWIDTH? -> :
CHANNEL11:STRAIN:BWIDTH 10.00E+00
```

:CHANnel<x>:STRain:EXCitation

Function Sets or queries the bridge voltage of an installed strain module.

Syntax
```
:CHANnel<x>:STRain:
EXCitation {<Voltage>}
:CHANnel<x>:STRain:EXCitation?
<Voltage> = 2V, 5V, 10V
```

Example
```
:CHANNEL11:STRAIN:EXCITATION 5
:CHANNEL11:STRAIN:EXCITATION? ->
:CHANNEL11:STRAIN:EXCITATION 5
```

:CHANnel<x>:STRain:GFACtor

Function Sets or queries the gauge factor of an installed strain module.

Syntax
```
:CHANnel<x>:STRain:GFACtor {<NRf>}
:CHANnel<x>:STRain:GFACtor?
<NRf> = 1.90 to 2.20
```

Example
```
:CHANNEL11:STRAIN:GFACTOR 1.9
:CHANNEL11:STRAIN:GFACTOR? ->
:CHANNEL11:STRAIN:GFACTOR 1.90
```

:CHANnel<x>:STRain:INVert

Function Sets or queries whether the display is inverted (ON) or not (OFF) for an installed strain module.

Syntax
```
:CHANnel<x>:STRain:
INVert {<Boolean>}
:CHANnel<x>:STRain:INVert?
```

Example
```
:CHANNEL11:STRAIN:INVERT ON
:CHANNEL11:STRAIN:INVERT? ->
:CHANNEL11:STRAIN:INVERT 1
```

:CHANnel<x>:STRain:LSCale?

Function Queries all the linear scaling settings of an installed strain module.

Syntax
```
:CHANnel<x>:STRain:LSCale?
```

:CHANnel<x>:STRain:LSCale:AVALue

Function Sets or queries the scaling coefficient A that is used during linear scaling for an installed strain module.

Syntax
```
:CHANnel<x>:STRain:LSCale:
AVALue {<NRf>}
:CHANnel<x>:STRain:LSCale:AVALue?
<x> = 1 to 16
<NRf> = -9.9999E+30 to -1E-30,
1E-30 to 9.9999E+30
```

Example
```
:CHANNEL1:STRAIN:LSCALE:
AVALUE 25.0000E+00
:CHANNEL1:STRAIN:LSCALE:AVALUE? ->
:CHANNEL1:STRAIN:LSCALE:
AVALUE 25.0000E+00
```

Description The coefficient cannot be set to 0.

:CHANnel<x>:STRain:LSCale:BVALue

Function Sets or queries the scaling coefficient B that is used during linear scaling for an installed strain module.

Syntax
```
:CHANnel<x>:STRain:LSCale:
BVALue {<NRf>}
:CHANnel<x>:STRain:LSCale:BVALue?
<x> = 1 to 16
<NRf> = -9.9999E+30 to -1E-30, 1E-30
to 9.9999E+30
```

Example
```
:CHANNEL1:STRAIN:LSCALE:
BVALUE 25.0000E+00
:CHANNEL1:STRAIN:LSCALE:BVALUE? ->
:CHANNEL1:STRAIN:LSCALE:
BVALUE 25.0000E+00
```

Description The coefficient cannot be set to 0.

:CHANnel<x>:STRain:LSCale:DISPlaytype?

Function Queries all the linear-scaling display-mode settings of an installed strain module.

Syntax
```
:CHANnel<x>:STRain:LSCale:
DISPlaytype?
```

:CHANnel<x>:STRain:LSCale:DISPlaytype:MODE

Function Sets or queries the display mode that is used during linear scaling for an installed strain module.

Syntax
```
:CHANnel<x>:STRain:LSCale:
DISPlaytype:MODE {EXPonent|FLOating}
:CHANnel<x>:STRain:LSCale:
DISPlaytype:MODE?
<x> = 1 to 16
```

Example
```
:CHANNEL1:STRAIN:LSCALE:DISPLAYTYPE:
MODE EXPONENT
:CHANNEL1:STRAIN:LSCALE:DISPLAYTYPE:
MODE? -> :CHANNEL1:STRAIN:LSCALE:
DISPLAYTYPE:MODE EXPONENT
```

:CHANnel<x>:STRain:LSCale:DISPlaytype:DECimalnum

Function Sets or queries the decimal place that is used during linear scaling when the display mode is set to Float for an installed strain module.

Syntax
```
:CHANnel<x>:STRain:LSCale:
DISPlaytype:DECimalnum {<NRf>|AUTO}
:CHANnel<x>:STRain:LSCale:
DISPlaytype:DECimalnum?
<x> = 1 to 16
<NRf> = 0 to 3
```

Example
```
:CHANNEL1:STRAIN:LSCALE:DISPLAYTYPE:
DECIMALNUM AUTO
:CHANNEL1:STRAIN:LSCALE:DISPLAYTYPE:
DECIMALNUM? -> :CHANNEL1:STRAIN:
LSCALE:DISPLAYTYPE:DECIMALNUM AUTO
```

:CHANnel<x>:STRain:LSCale:DISPlaytype:SUBunit

Function Sets or queries the unit prefix that is used during linear scaling when the display mode is set to Float for an installed strain module.

Syntax
```
:CHANnel<x>:STRain:LSCale:
DISPlaytype:SUBunit {AUTO|NONE|PICO|
NANO|MICRo|MILI|KILO|MEGA|GIGA|TERA}
:CHANnel<x>:STRain:LSCale:
DISPlaytype:SUBunit?
<x> = 1 to 16
```

Example
```
:CHANNEL1:STRAIN:LSCALE:DISPLAYTYPE:
SUBUNIT AUTO
:CHANNEL1:STRAIN:LSCALE:DISPLAYTYPE:
SUBUNIT? -> :CHANNEL1:STRAIN:LSCALE:
DISPLAYTYPE:SUBUNIT AUTO
```

:CHANnel<x>:STRain:LSCale:GETMeasure

Function Executes the measurement of the linear scaling P1X or P2X value of an installed strain module.

Syntax
```
:CHANnel<x>:STRain:LSCale:
GETMeasure {P1X|P2X}
<x> = 1 to 16
```

Example
```
:CHANnel1:STRAIN:LSCale:
GETMeasure P1X
```

:CHANnel<x>:STRain:LSCale:MODE

Function Sets or queries the linear scale mode of an installed strain module.

Syntax
```
:CHANnel<x>:STRain:LSCale:MODE {AXB|
OFF|P12|SHUNt}
:CHANnel<x>:STRain:LSCale:MODE?
SHUNt can only be set when the
installed module is a 701271.
```

Example
```
:CHANNEL11:STRAIN:LSCALE:MODE OFF
:CHANNEL11:STRAIN:LSCALE:MODE? ->
:CHANNEL11:STRAIN:LSCALE:MODE OFF
```

:CHANnel<x>:STRain:LSCale:{P1X|P1Y|P2X|P2Y}

Function Sets or queries the linear scaling P1X, P1Y, P2X, or P2Y value of an installed strain module.

Syntax
```
:CHANnel<x>:STRain:LSCale:{P1X|P1Y|
P2X|P2Y} {<NRf>}
:CHANnel<x>:STRain:LSCale:{P1X|P1Y|
P2X|P2Y}?
<x> = 1 to 16
P1X or P2X's <NRf> = -9.9999E+30 to
-1E-30, 0, 1E-30 to 9.9999E+30
P1Y or P2Y's <NRf> = -9.9999E+25 to
-1E-25, 0, 1E-25 to 9.9999E+25
```

Example
```
:CHANNEL1:STRAIN:LSCALE:
P1X 1.00000E+00
:CHANNEL1:STRAIN:LSCALE:P1X? ->
:CHANNEL1:STRAIN:LSCALE:P1X
1.00000E+00
```

:CHANnel<x>:STRain:LSCale:SHUNt

Function Executes shunt calibration.

Syntax
```
:CHANnel<x>:STRain:LSCale:SHUNt
<x> = 1 to 16
```

:CHANnel<x>:STRain:LSCale:UNIT

Function Sets or queries the linear-scaling unit string of an installed strain module.

Syntax
```
:CHANnel<x>:STRain:LSCale:
UNIT {<String>}
:CHANnel<x>:STRain:LSCale:UNIT?
<x> = 1 to 16
<String> = Up to 4 characters
```

Example
```
:CHANNEL1:STRAIN:LSCALE:UNIT "UU"
:CHANNEL1:STRAIN:LSCALE:UNIT? ->
:CHANNEL1:STRAIN:LSCALE:UNIT "UU"
```

:CHANnel<x>:STRain:RANGe

Function Sets or queries the measurement range of an installed strain module.

Syntax
```
:CHANnel<x>:STRain:RANGe {<NRf>}
:CHANnel<x>:STRain:RANGe?
<NRf> = 0.25, 0.5, 1, 2.5, 5, 10 (when
the unit is mV or V), 500, 1000,
2000, 5000, 10000, 20000 (when the
unit is µSTR)
```

Example
```
:CHANNEL11:STRAIN:RANGE 20000
:CHANNEL11:STRAIN:RANGE? ->
:CHANNEL11:STRAIN:RANGE 20000
```

:CHANnel<x>:STRain:SCALe

Function Sets or queries the upper and lower limits of the screen for an installed strain module.

Syntax
```
:CHANnel<x>:STRain:
SCALe {<NRf>,<NRf>}
:CHANnel<x>:STRain:SCALe?
<NRf> = -15.0000 to 15.0000 (when
the unit is mV or V)
-30000 to 30000 (when the unit is µSTR)
-9.9999E+30 to 9.9999E+30 (during
linear scaling)
```

Example
```
:CHANNEL5:STRAIN:SCALE 5000,-5000
:CHANNEL5:STRAIN:SCALE? ->
:CHANNEL5:STRAIN:SCALE
5.000000E+03,-5.000000E+03
```

:CHANnel<x>:STRain:UNIT

Function Sets or queries the unit of an installed strain module.

Syntax
```
:CHANnel<x>:STRain:UNIT {MV|USTR}
:CHANnel<x>:STRain:UNIT?
```

Example
```
:CHANNEL11:STRAIN:UNIT USTR
:CHANNEL11:STRAIN:UNIT? ->
:CHANNEL11:STRAIN:UNIT USTR
```

:CHANnel<x>:TEMPerature?

Function Queries all the settings of a module if the module in the specified channel can measure temperature.

Syntax
```
:CHANnel<x>:TEMPerature?
```

:CHANnel<x>:TEMPerature:BURNout

Function Sets or queries whether burnout will be detected when temperature measurements are performed.

Syntax
```
:CHANnel<x>:TEMPerature:
BURNout {<Boolean>}
:CHANnel<x>:TEMPerature:BURNout?
<x> = 1 to 16
```

Example
```
:CHANNEL9:TEMPERATURE:BURNOUT ON
:CHANNEL9:TEMPERATURE:BURNOUT? -> :
CHANNEL9:TEMPERATURE:BURNOUT 1
```

:CHANnel<x>:TEMPerature:BWIDth

Function Sets or queries the bandwidth limit that is used when temperature measurements are performed.

Syntax
```
:CHANnel<x>:TEMPerature:
BWIDth {AUTO|FULL|<Frequency>}
:CHANnel<x>:TEMPerature:BWIDth?
<x> = 1 to 16
For the 701265 {FULL|<Frequency>}
<Frequency> = 2, 8, 30 (Hz)
For the 701261 {FULL|<Frequency>}
<Frequency> = 2, 8, 30 (Hz)
For the 701262 {AUTO|FULL|
<Frequency>} <Frequency> = 2, 8, 30 (Hz)
```

Example
```
:CHANNEL9:TEMPERATURE:BWIDTH FULL
:CHANNEL9:TEMPERATURE:BWIDTH? ->
:CHANNEL9:TEMPERATURE:BWIDTH FULL
```

:CHANnel<x>:TEMPerature:COUPling

Function Sets or queries the input coupling setting of an installed module that can measure temperature.

Syntax
```
For the 701265
:CHANnel<x>:TEMPerature:
COUPling {DC|TC|GND}
For the 701261 or 701262
:CHANnel<x>:TEMPerature:
COUPling {AC|DC|TC|GND}
<x> = 1 to 16
:CHANnel<x>:TEMPerature:COUPling?
```

Example
```
:CHANNEL9:TEMPERATURE:COUPLING TC
:CHANNEL9:TEMPERATURE:COUPLING? ->
:CHANNEL9:TEMPERATURE:COUPLING TC
```

:CHANnel<x>:TEMPerature:RJC

Function Sets or queries the RJC that is used when temperature measurements are performed.

Syntax
```
:CHANnel<x>:TEMPerature:
RJC {<Boolean>}
:CHANnel<x>:TEMPerature:RJC?
<x> = 1 to 16
```

Example
```
:CHANNEL9:TEMPERATURE:RJC 1
:CHANNEL9:TEMPERATURE:RJC? ->
:CHANNEL9:TEMPERATURE:RJC 1
```

:CHANnel<x>:TEMPerature:SCALe

Function Sets or queries the upper and lower limits of the screen that are used when temperature measurements are performed.

Syntax
```
:CHANnel<x>:TEMPerature:
SCALe {<NRf>,<NRf>}
:CHANnel<x>:TEMPerature:SCALe?
<x> = 1 to 16
<NRf> = -5432 to 5432 (in 0.1 steps)
```

Example
```
:CHANNEL9:TEMPERATURE:
SCALE 1300.0,-200.0
:CHANNEL9:TEMPERATURE:SCALE? ->
:CHANNEL9:TEMPERATURE:
SCALE 1300.0,-200.0
```

:CHANnel<x>:TEMPerature:TYPE

Function Sets or queries the thermocouple type that is used when temperature measurements are performed.

Syntax
```
:CHANnel<x>:TEMPerature:TYPE {K|E|J|
T|L|U|N|R|S|B|W|Au7fe}
:CHANnel<x>:TEMPerature:TYPE?
<x> = 1 to 16
```

Example
```
:CHANNEL9:TEMPERATURE:TYPE K
:CHANNEL9:TEMPERATURE:TYPE? ->
:CHANNEL9:TEMPERATURE:TYPE K
```

:CHANnel<x>:TEMPerature:UNIT

Function Sets or queries the unit that is used when temperature measurements are performed.

Syntax
```
:CHANnel<x>:TEMPerature:UNIT {C|K}
:CHANnel<x>:TEMPerature:UNIT?
<x> = 1 to 16
```

Example
```
:CHANNEL9:TEMPERATURE:UNIT C
:CHANNEL9:TEMPERATURE:UNIT? ->
:CHANNEL9:TEMPERATURE:UNIT C
```

:CHANnel<x>:VOLTage?

Function Queries all settings of an installed module that can perform analog voltage measurements.

Syntax
```
:CHANnel<x>:VOLTage?
```

:CHANnel<x>[:VOLTage]:BWIDth

Function Sets or queries the bandwidth limit of an installed module that can perform analog voltage measurements.

Syntax
```
:CHANnel<x>[:VOLTage]:
BWIDth {FULL|<Frequency>}
:CHANnel<x>[:VOLTage]:BWIDth?
<x> = 1 to 16
```

Example
```
:CHANNEL2:VOLTAGE:BWIDTH FULL
:CHANNEL2:VOLTAGE:BWIDTH? ->
:CHANNEL2:VOLTAGE:BWIDTH FULL
```

Description

List of Frequency Settings per Voltage Module Type

Type	List of Frequency Settings	Notes
701250/55	500Hz, 5kHz, 50kHz, 500kHz	
701251	400Hz, 4kHz, 40kHz	
701260	100Hz, 1kHz, 10kHz	
701261	40Hz, 400Hz, 4kHz	When the input coupling is set to a setting other than TC
	2Hz, 8Hz, 30Hz	When the input coupling is set to TC
701262	40Hz, 400Hz, 4kHz	When the input coupling is set to a setting other than TC
	AUTO, 2Hz, 8Hz, 30Hz	When the input coupling is set to TC
701265	2Hz, 8Hz, 30Hz	
701275	40Hz, 400Hz, 4kHz, AUTO	
720210	10kHz, 20kHz, 40kHz, 80kHz, 160kHz, 640kHz, 1.28MHz, 2MHz	

:CHANnel<x>[:VOLTage]:COUPling

Function Sets or queries the input coupling of an installed module that can perform analog voltage measurements.

Syntax
```
:CHANnel<x>[:VOLTage]:COUPling {AC|
DC|GND|ACRMS|DCRMS|TC|ACCL}
:CHANnel<x>[:VOLTage]:COUPling?
<x> = 1 to 16
```

Example
```
:CHANNEL2:VOLTAGE:COUPLING DC
:CHANNEL2:VOLTAGE:COUPLING? ->
:CHANNEL2:VOLTAGE:COUPLING DC
```

Description For 701250, 701251, 701255, or 720210: {AC|DC|GND}
For 701260: {AC|DC|GND|ACRMS|DCRMS}
For 701261 or 701262: {AC|DC|GND|TC}
For 701265: {DC|GND|TC}
For 701275: {AC|DC|GND|ACCL}

:CHANnel<x>[:VOLTage]:INVert

Function Sets or queries whether the display is inverted (ON) or not (OFF) for an installed module that can perform analog voltage measurements.

Syntax
```
:CHANnel<x>[:VOLTage]:
INVert {<Boolean>}
:CHANnel<x>[:VOLTage]:INVert?
<x> = 1 to 16
```

Example
```
:CHANNEL2:VOLTAGE:INVERT ON
:CHANNEL2:VOLTAGE:INVERT? ->
:CHANNEL2:VOLTAGE:INVERT 1
```

:CHANnel<x>[:VOLTage]:LSCale?

Function Queries all the linear scaling settings of an installed module that can perform analog voltage measurements.

Syntax
```
:CHANnel<x>[:VOLTage]:LSCale?
```

:CHANnel<x>[:VOLTage]:LSCale:AVALue

Function Sets or queries the scaling coefficient A that is used during linear scaling for an installed module that can perform analog voltage measurements.

Syntax
```
:CHANnel<x>[:VOLTage]:LSCale:
AVALue {<NRf>}
:CHANnel<x>[:VOLTage]:LSCale:AVALue?
<x> = 1 to 16
<NRf> = -9.9999E+30 to -1E-30, 1E-30
to 9.9999E+30
```

Example
```
:CHANNEL2:VOLTAGE:LSCALE:
AVALUE 25.0000E+00
:CHANNEL2:VOLTAGE:LSCALE:AVALUE? ->
:CHANNEL2:VOLTAGE:LSCALE:
AVALUE 25.0000E+00
```

Description The coefficient cannot be set to 0.

:CHANnel<x>[:VOLTage]:LSCale:BVALue

Function Sets or queries the offset value B that is used during linear scaling for an installed module that can perform analog voltage measurements.

Syntax
```
:CHANnel<x>[:VOLTage]:LSCale:
BVALue {<NRf>}
:CHANnel<x>[:VOLTage]:LSCale:BVALue?
<x> = 1 to 16
<NRf> = -9.9999E+30 to -1E-30, 0,
1E-30 to 9.9999E+30
```

Example
```
:CHANNEL2:VOLTAGE:LSCALE:
BVALUE -25.0000E+00
:CHANNEL2:VOLTAGE:LSCALE:BVALUE? ->
:CHANNEL2:VOLTAGE:LSCALE:
BVALUE 25.0000E+00
```

:CHANnel<x>[:VOLTage]:LSCale:DISPlaytype?

Function Queries all the linear-scaling display-mode settings of an installed module that can perform analog voltage measurements.

Syntax
```
:CHANnel<x>[:VOLTage]:LSCale:
DISPlaytype?
```

:CHANnel<x>[:VOLTage]:LSCale:DISPlaytype:MODE

Function Sets or queries the display mode that is used during linear scaling for an installed module that can perform analog voltage measurements.

Syntax
```
:CHANnel<x>[:VOLTage]:LSCale:
DISPlaytype:MODE {EXPonent|FLOating}
:CHANnel<x>[:VOLTage]:LSCale:
DISPlaytype:MODE?
<x> = 1 to 16
```

Example
```
:CHANNEL1:VOLTAGE:LSCALE:
DISPLAYTYPE:MODE EXPONENT
:CHANNEL1:VOLTAGE:LSCALE:
DISPLAYTYPE:MODE? -> :CHANNEL1
:VOLTAGE:LSCALE:DISPLAYTYPE:
MODE EXPONENT
```

:CHANnel<x>[:VOLTage]:LSCale:DISPlaytype:DECimalnum

Function Sets or queries the decimal place that is used during linear scaling when the display mode is set to Float for an installed module that can perform analog voltage measurements.

Syntax
```
:CHANnel<x>[:VOLTage]:LSCale:
DISPlaytype:DECimalnum {<NRf>|AUTO}
:CHANnel<x>[:VOLTage]:LSCale:
DISPlaytype:DECimalnum?
<x> = 1 to 16
<NRf> = 0 to 3
```

Example
```
:CHANNEL1:VOLTAGE:LSCALE:
DISPLAYTYPE:DECIMALNUM AUTO
:CHANNEL1:VOLTAGE:LSCALE:
DISPLAYTYPE:DECIMALNUM? ->
:CHANNEL1:VOLTAGE:LSCALE:
DISPLAYTYPE:DECIMALNUM AUTO
```

:CHANnel<x>[:VOLTage]:LSCale:DISPlaytype:SUBunit

Function Sets or queries the unit prefix that is used during linear scaling when the display mode is set to Float for an installed module that can perform analog voltage measurements.

Syntax
```
:CHANnel<x>[:VOLTage]:LSCale:
DISPlaytype:SUBunit {AUTO|NONE|PICO|
NANO|MICRo|MILI|KILO|MEGA|GIGA|TERA}
:CHANnel<x>[:VOLTage]:LSCale:
DISPlaytype:SUBunit?
<x> = 1 to 16
```

Example
```
:CHANNEL1:VOLTAGE:LSCALE:
DISPLAYTYPE:SUBUNIT AUTO
:CHANNEL1:VOLTAGE:LSCALE:
DISPLAYTYPE:SUBUNIT? -> :CHANNEL1
:VOLTAGE:LSCALE:DISPLAYTYPE:
SUBUNIT AUTO
```

:CHANnel<x>[:VOLTage]:LSCale:GETMeasure

Function Executes the measurement of the linear scaling P1X or P2X value of an installed module that can perform analog voltage measurements.

Syntax
```
:CHANnel<x>[:VOLTage]:LSCale:
GETMeasure {P1X|P2X}
<x> = 1 to 16
```

:CHANnel<x>[:VOLTage]:LSCale:MODE

Function Sets or queries the linear scale mode of an installed module that can perform analog voltage measurements.

Syntax
```
:CHANnel<x>[:VOLTage]:LSCale:
MODE {AXB|OFF|P12}
:CHANnel<x>[:VOLTage]:LSCale:MODE?
<x> = 1 to 16
```

Example
```
:CHANNEL1:VOLTAGE:LSCALE:MODE AXB
:CHANNEL1:VOLTAGE:LSCALE:MODE? ->
:CHANNEL1:VOLTAGE:LSCALE:MODE OFF
```

:CHANnel<x>[:VOLTage]:LSCale:{P1X|P1Y|P2X|P2Y}

Function Sets or queries the linear scaling P1X, P1Y, P2X, or P2Y value of an installed module that can perform analog voltage measurements.

Syntax
```
:CHANnel<x>[:VOLTage]:
LSCale:{P1X|P1Y|P2X|P2Y} {<NRf>}
:CHANnel<x>[:VOLTage]:LSCale:{P1X|
P1Y|P2X|P2Y}?
<x> = 1 to 16
P1X or P2X's <NRf> = -9.9999E+30 to
-1E-30, 0, 1E-30 to 9.9999E+30
P1Y or P2Y's <NRf> = -9.9999E+25 to
-1E-25, 0, 1E-25 to 9.9999E+25
```

Example
```
:CHANNEL1:VOLTAGE:LSCALE:P1X 1
:CHANNEL1:VOLTAGE:LSCALE:P1X? ->
:CHANNEL1:VOLTAGE:LSCALE:P1X 1
```

:CHANnel<x>[:VOLTage]:LSCale:UNIT

Function Sets or queries the linear-scaling unit string of an installed module that can perform analog voltage measurements.

Syntax
```
:CHANnel<x>[:VOLTage]:LSCale:
UNIT {<String>}
:CHANnel<x>[:VOLTage]:LSCale:UNIT?
<x> = 1 to 16
<String> = Up to 4 characters
```

Example
```
:CHANNEL1:VOLTAGE:LSCALE:UNIT "UU"
:CHANNEL1:VOLTAGE:LSCALE:UNIT? ->
:CHANNEL1:VOLTAGE:LSCALE:UNIT "UU"
```

:CHANnel<x>[:VOLTage]:OFFSet

Function Sets or queries the offset voltage of an installed module that can perform analog voltage measurements.

Syntax
```
:CHANnel<x>[:VOLTage]:
OFFSet {<Voltage>|<Current>}
:CHANnel<x>[:VOLTage]:OFFSet?
<x> = 1 to 16
<Voltage> and <Current> = The
selectable range varies depending on
the range setting.
```

Example
```
:CHANNEL1:VOLTAGE:OFFSET 0
:CHANNEL1:VOLTAGE:OFFSET? ->
:CHANNEL1:VOLTAGE:OFFSET 1.00000E+00
```

:CHANnel<x>[:VOLTage]:POSition

Function Sets or queries the vertical position of an installed module that can perform analog voltage measurements.

Syntax
```
:CHANnel<x>[:VOLTage]:
POSition {<NRf>}
:CHANnel<x>[:VOLTage]:POSition?
<x> = 1 to 16
<NRf> = -5.00 to 5.00 (div; in 0.01
div steps)
```

Example
```
:CHANNEL1:VOLTAGE:POSITION 1.00
:CHANNEL1:VOLTAGE:POSITION? ->
:CHANNEL1:VOLTAGE:POSITION 1.00
```

:CHANnel<x>[:VOLTage]:PROBe

Function Sets or queries the probe type of an installed module that can perform analog voltage measurements.

Syntax
```
:CHANnel<x>[:VOLTage]:PROBe {<NRf>|
C10|C100}
:CHANnel<x>[:VOLTage]:PROBe?
<x> = 1 to 16
<NRf> = 1, 10, 100, 1000
```

Example
```
:CHANNEL1:VOLTAGE:PROBE 10
:CHANNEL1:VOLTAGE:PROBE? ->
:CHANNEL1:VOLTAGE:PROBE 10
```

Description You cannot set or query the probe type for the 701261, 701262, or 701265.

:CHANnel<x>[:VOLTage]:SCALe

Function Sets or queries the upper and lower limits of the screen for an installed module that can perform analog voltage measurements.

Syntax
```
:CHANnel<x>[:VOLTage]:SCALe {<Voltage>,
<Voltage>|<Current>,<Current>|
<NRf>,<NRf>}
:CHANnel<x>[:VOLTage]:SCALe?
<x> = 1 to 16
<Voltage>, <Current>, and <NRf> = The
selectable range varies depending on
the range setting.
```

Example
```
:CHANNEL1:VOLTAGE:SCALE 250,-250
:CHANNEL1:VOLTAGE:SCALE? ->
:CHANNEL1:VOLTAGE:
SCALE 250.000E+00,-250.000E+00
```

:CHANnel<x>[:VOLTage]:VARiable

Function Sets or queries how the vertical scale is set for an installed module that can perform analog voltage measurements.

Syntax
```
:CHANnel<x>[:VOLTage]:
VARiable {<Boolean>}
:CHANnel<x>[:VOLTage]:VARiable?
<x> = 1 to 16
```

Example
```
:CHANNEL1:VOLTAGE:VARIABLE 0
:CHANNEL1:VOLTAGE:VARIABLE? ->
:CHANNEL1:VOLTAGE:VARIABLE 0
```

Description On the DL850/DL850V, the menu title is “V Scale.” OFF corresponds to the “DIV” setting and ON corresponds to the “SPAN” setting.

:CHANnel<x>[:VOLTage]:VDIV

Function Sets or queries the V/div setting of an installed module that can perform analog voltage measurements.

Syntax
```
:CHANnel<x>[:VOLTage]:
VDIV {<Voltage>|<Current>}
:CHANnel<x>[:VOLTage]:VDIV?
<x> = 1 to 16
<Voltage> = 10mV/div to 20V/div
(720210, when the probe attenuation
is 1:1)
```

Example
```
:CHANNEL1:VOLTAGE:VDIV 50
:CHANNEL1:VOLTAGE:VDIV? -> :CHANNEL1
:VOLTAGE:VDIV 50
```

Description

List of Selectable Ranges and Initial Values per Voltage Module Type (1:1 probe attenuation)

Type	Selectable Range of VDIV	Initial Value of VDIV	Selectable Range of the Offset
701250/55	5 mV to 20 V	5 V	The VDIV setting × ±5
701251	1 mV to 20 V	5 V	
701260	20 mV to 200 V	5 V	
701261/62	5 mV to 20 V	5 V	
701265	0.1 mV to 10 V	5 V	
701275	5 mV to 10 V	5 V	
720210	10 mV to 20 V	5 V	

:CHANnel<x>[:VOLTage]:ZOOM

Function Sets or queries the vertical zoom factor of an installed module that can perform analog voltage measurements.

Syntax
```
:CHANnel<x>[:VOLTage]:ZOOM {<NRf>}
:CHANnel<x>[:VOLTage]:ZOOM?
<x> = 1 to 16
<NRf> = 0.1, 0.111, 0.125, 0.143,
0.167, 0.2, 0.25, 0.33, 0.4, 0.5,
0.556, 0.625, 0.667, 0.714, 0.8,
0.833, 1, 1.11, 1.25, 1.33, 1.43, 1.67,
2, 2.22, 2.5, 3.33, 4, 5, 6.67, 8,
10, 12.5, 16.7, 20, 25, 40, 50, 100
However, the 701260 has the following
limits:
When using 1:1 probe attenuation and
50 V/div    0.25 to 100
When using 1:1 probe attenuation and
100 V/div   0.5 to 100
When using 1:1 probe attenuation and
200 V/div   1 to 100
```

Example
```
:CHANNEL1:VOLTAGE:ZOOM 1
:CHANNEL1:VOLTAGE:ZOOM? -> :CHANNEL1
:VOLTAGE:ZOOM 1
```

:CHANnel<x>[:VOLTage]:SCHannel<x>?

Function Queries all settings of an installed 16-CH Voltage Input Module.

Syntax
```
:CHANnel<x>[:VOLTage]:SCHannel<x>?
```

:CHANnel<x>[:VOLTage]:SCHannel<x>:LABel

Function Sets or queries the label string of the specified sub channel for the specified 16-CH Voltage Input Module.

Syntax
```
:CHANnel<x>[:VOLTage]:SCHannel<x>:
LABel {<String>}
:CHANnel<x>[:VOLTage]:SCHannel<x>:
LABel?
CHANnel<x>'s <x> = 1 to 16
SCHannel<x>'s <x> = 1 to 16 (the
upper limit varies depending on the
module type)
<String> = Up to 16 characters
```

:CHANnel<x>[:VOLTage]:SCHannel<x>:BWIDth

Function Sets or queries the bandwidth limit of an installed 16-CH Voltage Input Module.

Syntax
```
:CHANnel<x>[:VOLTage]:SCHannel<x>:
BWIDth {FULL|<Frequency>}
:CHANnel<x>[:VOLTage]:SCHannel<x>:
BWIDth?
CHANnel<x>'s <x> = 1 to 16 (only
odd-numbered channels are valid)
SCHannel<x>'s <x> = 1 to 16
<Frequency> = 500Hz
```

Description Error 113, “Undefined header,” will occur if a module other than a 16-CH Voltage Input Module is installed in the slot that corresponds to the specified channel.

:CHANnel<x>[:VOLTage]:SCHannel<x>:COUPling

Function Sets or queries the input coupling setting of an installed 16-CH Voltage Input Module.

Syntax
```
:CHANnel<x>[:VOLTage]:SCHannel<x>:
COUPling {DC|GND|OFF}
:CHANnel<x>[:VOLTage]:SCHannel<x>:
COUPling?
CHANnel<x>'s <x> = 1 to 16 (only
odd-numbered channels are valid)
SCHannel<x>'s <x> = 1 to 16
```

Description Error 113, “Undefined header,” will occur if a module other than a 16-CH Voltage Input Module is installed in the slot that corresponds to the specified channel.
You cannot use this command or menu operations to change the setting while measurements are being performed.

:CHANnel<x>[:VOLTage]:SCHannel<x>:INVert

Function Sets or queries whether the display is inverted (ON) or not (OFF) for an installed 16-CH Voltage Input Module.

Syntax
```
:CHANnel<x>[:VOLTage]:SCHannel<x>:
INVert {<Boolean>}
:CHANnel<x>[:VOLTage]:SCHannel<x>:
INVert?
CHANnel<x>'s <x> = 1 to 16 (only
odd-numbered channels are valid)
SCHannel<x>'s <x> = 1 to 16
```

Description Error 113, “Undefined header,” will occur if a module other than a 16-CH Voltage Input Module is installed in the slot that corresponds to the specified channel.

:CHANnel<x1>[:VOLTage]:SCHannel<x2>:LSCale?

Function Queries all the linear scaling settings of an installed 16-CH Voltage Input Module.

Syntax
```
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
LSCale?
```

:CHANnel<x1>[:VOLTage]:SCHannel<x2>:LSCale:AVALue

Function Sets or queries the scaling coefficient A that is used during linear scaling for an installed 16-CH Voltage Input Module.

Syntax
```
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
LSCale:AVALue {<NRf>}
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
LSCale:AVALue?
<x1> = 1, 3, 5, 7, 9, 11, 13, 15
<x2> = 1 to 16
<NRf> = -9.9999E+30 to -1E-30, 1E-30
to 9.9999E+30
```

Example
```
:CHANNEL3:VOLTAGE:SCHANNEL1:LSCALE:
AVALUE 25
:CHANNEL3:VOLTAGE:SCHANNEL1:LSCALE:
AVALUE? -> :CHANNEL3:VOLTAGE
:SCHANNEL1:LSCALE:AVALUE 25.0000E+00
```

Description The coefficient cannot be set to 0.

:CHANnel<x1>[:VOLTage]:SCHannel<x2>:LSCale:BVALue

Function Sets or queries the offset value B that is used during linear scaling for an installed 16-CH Voltage Input Module.

Syntax
```
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
LSCale:BVALue {<NRf>}
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
LSCale:BVALue?
<x1> = 1, 3, 5, 7, 9, 11, 13, 15
<x2> = 1 to 16
<NRf> = -9.9999E+30 to -1E-30, 0, 1E
-30 to 9.9999E+30
```

Example
```
:CHANNEL3:VOLTAGE:SCHANNEL1:LSCALE
:BVALUE -25
:CHANNEL3:VOLTAGE:SCHANNEL1:LSCALE
:BVALUE? -> :CHANNEL3:VOLTAGE
:SCHANNEL1:LSCALE:
BVALUE -25.0000E+00
```

:CHANnel<x1>[:VOLTage]:SCHannel<x2>:LSCale:DISPlaytype?

Function Queries all the linear-scaling display-mode settings of an installed 16-CH Voltage Input Module.

Syntax
```
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
LSCale:DISPlaytype?
```

:CHANnel<x1>[:VOLTage]:SCHannel<x2>:LSCale:DISPlaytype:MODE

Function Sets or queries the display mode that is used during linear scaling for an installed 16-CH Voltage Input Module.

Syntax
```
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
LSCale:DISPlaytype:MODE
{EXPonent|FLOating}
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
LSCale:DISPlaytype:MODE?
<x1> = 1, 3, 5, 7, 9, 11, 13, 15
<x2> = 1 to 16
```

Example
```
:CHANNEL3:VOLTAGE:SCHANNEL1:LSCALE:
DISPLAYTYPE:MODE EXPONENT
:CHANNEL3:VOLTAGE:SCHANNEL1:LSCALE:
DISPLAYTYPE:MODE? -> :CHANNEL1
:VOLTAGE:LSCALE:DISPLAYTYPE:
MODE EXPONENT
```

:CHANnel<x1>[:VOLTage]:SCHannel<x2>:LSCale:DISPlaytype:DECimalnum

Function Sets or queries the decimal place that is used during linear scaling when the display mode is set to Float for an installed 16-CH Voltage Input Module.

Syntax
```
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
LSCale:DISPlaytype:
DECimalnum {<NRf>|AUTO}
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
LSCale:DISPlaytype:DECimalnum?
<x1> = 1, 3, 5, 7, 9, 11, 13, 15
<x2> = 1 to 16
<NRf> = 0 to 3
```

Example
```
:CHANNEL3:VOLTAGE:SCHANNEL1:LSCALE
:DISPLAYTYPE:DECIMALNUM AUTO
:CHANNEL3:VOLTAGE:SCHANNEL1:LSCALE
:DISPLAYTYPE:DECIMALNUM? ->
:CHANNEL1:VOLTAGE:SCHANNEL1:LSCALE
:DISPLAYTYPE:DECIMALNUM AUTO
```

:CHANnel<x1>[:VOLTage]:SCHannel<x2>:LSCale:DISPlaytype:SUBunit

Function Sets or queries the unit prefix that is used during linear scaling when the display mode is set to Float for an installed 16-CH Voltage Input Module.

Syntax
```
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
LSCale:DISPlaytype:SUBunit {AUTO|
NONE|PICO|NANO|MICRo|MILI|KILO|MEGA|
GIGA|TERA}
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
LSCale:DISPlaytype:SUBunit?
<x1> = 1, 3, 5, 7, 9, 11, 13, 15
<x2> = 1 to 16
```

Example
```
:CHANNEL3:VOLTAGE:SCHANNEL1:LSCALE
:DISPLAYTYPE:SUBUNIT AUTO
:CHANNEL3:VOLTAGE:SCHANNEL1:LSCALE:
DISPLAYTYPE:SUBUNIT? -> :CHANNEL3
:VOLTAGE:SCHANNEL1:LSCALE
:DISPLAYTYPE:SUBUNIT AUTO
```

:CHANnel<x1>[:VOLTage]:SCHannel<x2>:LSCale:GETMeasure

Function Executes the measurement of the linear scaling P1X or P2X value of an installed 16-CH Voltage Input Module.

Syntax
```
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
LSCale:GETMeasure {P1X|P2X}
<x1> = 1, 3, 5, 7, 9, 11, 13, 15
<x2> = 1 to 16
```

:CHANnel<x1>[:VOLTage]:SCHannel<x2>:LSCale:MODE

Function Sets or queries the linear scale mode of an installed 16-CH Voltage Input Module.

Syntax
```
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
LSCale:MODE {AXB|OFF|P12}
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
LSCale:MODE?
<x1> = 1, 3, 5, 7, 9, 11, 13, 15
<x2> = 1 to 16
```

Example
```
:CHANNEL3:VOLTAGE:SCHANNEL1:LSCALE
:MODE AXB
:CHANNEL3:VOLTAGE:SCHANNEL1:LSCALE
:MODE? -> :CHANNEL3:VOLTAGE
:SCHANNEL1:LSCALE:MODE OFF
```

:CHANnel<x1>[:VOLTage]:SCHannel<x2>:LSCale:{P1X|P1Y|P2X|P2Y}

Function Sets or queries the linear scaling P1X, P1Y, P2X, or P2Y value of an installed 16-CH Voltage Input Module.

Syntax
```
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
LSCale:{P1X|P1Y|P2X|P2Y} {<NRf>}
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
LSCale:{P1X|P1Y|P2X|P2Y}?
<x1> = 1, 3, 5, 7, 9, 11, 13, 15
<x2> = 1 to 16
P1X or P2X's <NRf> = -9.9999E+30 to
-1E-30, 0, 1E-30 to 9.9999E+30
P1Y or P2Y's <NRf> = -9.9999E+25 to
-1E-25, 0, 1E-25 to 9.9999E+25
```

Example
```
:CHANNEL3:VOLTAGE:SCHANNEL1:LSCALE:
P1X 1
:CHANNEL3:VOLTAGE: SCHANNEL1:LSCALE:
P1X? -> :CHANNEL1:VOLTAGE:LSCALE:
P1X 1
```

:CHANnel<x1>[:VOLTage]:SCHannel<x2>:LSCale:UNIT

Function Sets or queries the linear-scaling unit string of an installed 16-CH Voltage Input Module.

Syntax
```
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
LSCale:UNIT {<String>}
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
LSCale:UNIT?
<x1> = 1, 3, 5, 7, 9, 11, 13, 15
<x2> = 1 to 16
<String> = Up to 4 characters
```

Example
```
:CHANNEL1:VOLTAGE:LSCALE:UNIT "UU"
:CHANNEL1:VOLTAGE:LSCALE:UNIT? ->
:CHANNEL1:VOLTAGE:LSCALE:UNIT "UU"
```

:CHANnel<x1>[:VOLTage]:SCHannel<x2>:OFFSet

Function Sets or queries the offset voltage of an installed 16-CH Voltage Input Module.

Syntax
```
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
OFFSet {<Voltage>}
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
OFFSet?
<x1> = 1 to 16 (only odd-numbered
channels are valid)
<x2> = 1 to 16
<Voltage> = The selectable range
varies depending on the range
setting.
```

Description Error 113, “Undefined header,” will occur if a module other than a 16-CH Voltage Input Module is installed in the slot that corresponds to the specified channel.

:CHANnel<x1>[:VOLTage]:SCHannel<x2>:POSition

Function Sets or queries the vertical position for an installed 16-CH Voltage Input Module.

Syntax
```
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
POSition {<NRf>}
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
POSition?
<x1> = 1 to 16 (only odd-numbered
channels are valid)
<x2> = 1 to 16
<NRf> = -5.00 to 5.00 (div; in 0.01
div steps)
```

Description Error 113, “Undefined header,” will occur if a module other than a 16-CH Voltage Input Module is installed in the slot that corresponds to the specified channel.

:CHANnel<x1>[:VOLTage]:SCHannel<x2>:SCALe

Function Sets or queries the upper and lower limits of the screen for an installed 16-CH Voltage Input Module.

Syntax
```
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
SCALe {<Voltage>,<Voltage>|
<NRf>,<NRf>}
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
SCALe?
<x1> = 1 to 16 (only odd-numbered
channels are valid)
<x2> = 1 to 16
<Voltage> and <NRf> = The selectable
range varies depending on the range
setting.
```

Description Error 113, “Undefined header,” will occur if a module other than a 16-CH Voltage Input Module is installed in the slot that corresponds to the specified channel.

:CHANnel<x1>[:VOLTage]:SCHannel<x2>:VARiable

Function Sets or queries how the vertical scale is set for an installed 16-CH Voltage Input Module.

Syntax
```
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
VARiable {<Boolean>}
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
VARiable?
<x1> = 1 to 16 (only odd-numbered
channels are valid)
<x2> = 1 to 16
```

Description Error 113, “Undefined header,” will occur if a module other than a 16-CH Voltage Input Module is installed in the slot that corresponds to the specified channel.

:CHANnel<x1>[:VOLTage]:SCHannel<x2>:VDIV

Function Sets or queries the V/DIV setting of an installed 16-CH Voltage Input Module.

Syntax
```
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
VDIV {<Voltage>}
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
VDIV?
<x1> = 1 to 16 (only odd-numbered
channels are valid)
<x2> = 1 to 16
<Voltage> = 200mV to 2V
```

Description Error 113, “Undefined header,” will occur if a module other than a 16-CH Voltage Input Module is installed in the slot that corresponds to the specified channel.

:CHANnel<x1>[:VOLTage]:SCHannel<x2>:ZOOM

Function Sets or queries the vertical zoom factor of an installed 16-CH Voltage Input Module.

Syntax
```
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
ZOOM {<NRf>}
:CHANnel<x1>[:VOLTage]:SCHannel<x2>:
ZOOM?
<x1> = 1 to 16 (only odd-numbered
channels are valid)
<x2> = 1 to 16
<NRf> = 0.1, 0.111, 0.125, 0.143,
0.167, 0.2, 0.25, 0.33, 0.4, 0.5,
0.556, 0.625, 0.667, 0.714, 0.8,
0.833, 1, 1.11, 1.25, 1.33, 1.43,
1.67, 2, 2.22, 2.5, 3.33, 4, 5, 6.67,
8, 10, 12.5, 16.7, 20, 25, 40, 50,
100
```

Description Error 113, “Undefined header,” will occur if a module other than a 16-CH Voltage Input Module is installed in the slot that corresponds to the specified channel.

5.7 CLEar Group

The command in this group deals with executing the clear trace operation. You can execute the same operation that you can execute by using the CLEAR TRACE key on the front panel.

:CLEar

Function Clears traces.

Syntax :CLEar

5.8 COMMunicate Group

The commands in this group deal with communications. There are no front panel keys that correspond to the commands in this group.

:COMMunicate?

Function Queries all communication settings.

Syntax
```
:COMMunicate?
```

:COMMunicate:HEADer

Function Sets or queries whether headers are attached to query responses.

Syntax
```
:COMMunicate:HEADer {<Boolean>}
:COMMunicate:HEADer?
```

Example
```
:COMMUNICATE:HEADER ON
:COMMUNICATE:HEADER? -> :COMMUNICATE:
HEADER 1
```

:COMMunicate:LOCKout

Function Sets or clears local lockout.

Syntax
```
:COMMunicate:LOCKout {<Boolean>}
:COMMunicate:LOCKout?
```

Example
```
:COMMUNICATE:LOCKOUT ON
:COMMUNICATE:LOCKOUT? ->
:COMMUNICATE:LOCKOUT 1
```

Description This command is designed for use in USB and Ethernet communications.

:COMMunicate:OPSE

Function Sets or queries the overlap command that is used by the *OPC, *OPC?, and *WAI commands.

Syntax
```
:COMMunicate:OPSE <Register>
:COMMunicate:OPSE?
<Register> = 0 to 65535
See the figure in the description of
the :COMM:WAIT? command.
```

Example
```
:COMMUNICATE:OPSE 65535
:COMMUNICATE:OPSE? -> :COMMUNICATE
:OPSE 584
```

Description Bits fixed to 0 are not set to 1, so the response to the query only indicates 1 for bits 5, 6, 8, and 11.

:COMMunicate:OPSR?

Function Queries the operation pending status register.

Syntax
```
:COMMunicate:OPSR?
```

Example
```
:COMMunicate:OPSR? -> 0
```

:COMMunicate:OVERlap

Function Sets or queries the commands that operate as overlap commands.

Syntax
```
:COMMunicate:OVERlap <Register>
:COMMunicate:OVERlap?
<Register> = 0 to 65535
See the figure in the description of
the :COMM:WAIT? command.
```

Description Bits fixed to 0 are not set to 1, so the response to the query only indicates 1 for bits 5, 6, 8, and 11.

:COMMunicate:REMote

Function Sets or queries whether the DL850/DL850V is in remote or local mode. ON is remote mode.

Syntax
```
:COMMunicate:REMote {<Boolean>}
:COMMunicate:REMote?
```

Example
```
:COMMUNICATE:REMOTE ON
:COMMUNICATE:REMOTE? -> :COMMUNICATE:
REMOTE 1
```

Description This command is designed for use in USB and Ethernet communications.

:COMMunicate:VERBose

Function Sets or queries whether query responses are returned in full or abbreviated form.

Syntax
```
:COMMunicate:VERBose {<Boolean>}
:COMMunicate:VERBose?
```

Example
```
:COMMUNICATE:VERBOSE ON
:COMMUNICATE:VERBOSE? ->
:COMMUNICATE:VERBOSE 1
```

Description This setting is lost when the DL850/DL850V is turned off.

:COMMunicate:WAIT

Function Waits for a specified extended event to occur.

Syntax
```
:COMMunicate:WAIT <Register>
<Register> = 0 to 65535 (extended
event register)
```

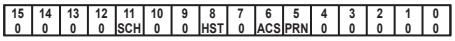
Example
```
:COMMUNICATE:WAIT 65535
```

:COMMunicate:WAIT?

Function Creates the response that is returned when a specified extended event occurs.

Syntax :COMMunicate:WAIT? <Register>
<Register> = 0 to 65535 (extended event register)

Example :COMMUNICATE:WAIT? 65535 -> 1



When bit 5 (PRN) = 1:
Printer operation is incomplete
When bit 6 (ACS) = 1:
Media access is incomplete
When bit 8 (HST) = 1:
History searching is incomplete
When bit 11 (SCH) = 1:
Edge searching is incomplete

5.9 CURSor Group

The commands in this group deal with cursor measurements. You can make the same settings and queries (of settings and measured values) that you can by using the CURSOR key on the front panel.
If the selectable range of the time axis is "<NRf> = –5 to 5div," the selectable range varies depending on settings such as the record length. For details, see "Selectable Range of Cursor Positions" under "Notes about Cursor Measurement" in chapter 9 of IM DL850-01EN.

:CURSor?
Function Queries all cursor measurement settings.
Syntax :CURSor?

:CURSor:FFT?
Function Queries all FFT cursor settings.
Syntax :CURSor:FFT?

:CURSor:FFT:MARKer:M<x>?
Function Queries all settings of an FFT cursor (marker cursor).
Syntax :CURSor:FFT:MARKer:M<x>?
<x> = 1 to 4

:CURSor:FFT:MARKer:M<x>:DF<y>?
Function Queries all ΔF display settings. ΔF is between the FFT cursors (marker cursors).
<x> = 1 to 4, <y> = 1 to 4

:CURSor:FFT:MARKer:M<x>:DF<y>:STATe
Function Sets or queries whether the ΔF value between the FFT cursors (marker cursors) is displayed.
Syntax :CURSor:FFT:MARKer:M<x>:DF<y>:
STATe {<Boolean>}
:CURSor:FFT:MARKer:M<x>:DF<y>:STATe?
<x> = 1 to 4, <y> = 1 to 4
Example :CURSOR:FFT:MARKER:M1:DF2:STATE ON
:CURSOR:FFT:MARKER:M1:DF2:STATE? ->
:CURSOR:FFT:MARKER:M1:DF2:STATE 1

:CURSor:FFT:MARKer:M<x>:DF<y>:VALue?
Function Queries the ΔF value between the FFT cursors (marker cursors).
Syntax :CURSor:FFT:MARKer:M<x>:DF<y>:VALue?
<x> = 1 to 4, <y> = 1 to 4
Example :CURSOR:FFT:MARKER:M1:DF2:VALUE? ->
:CURSOR:FFT:MARKER:M1:DF2:VALUE 100

:CURSor:FFT:MARKer:M<x>:DY<y>?
Function Queries all ΔY display settings. ΔY is between the FFT cursors (marker cursors).
Syntax :CURSor:FFT:MARKer:M<x>:DY<y>?
<x> = 1 to 4, <y> = 1 to 4

:CURSor:FFT:MARKer:M<x>:DY<y>:STATe
Function Sets or queries whether the ΔY value between the FFT cursors (marker cursors) is displayed.
Syntax :CURSor:FFT:MARKer:M<x>:DY<y>:
STATe {<Boolean>}
:CURSor:FFT:MARKer:M<x>:DY<y>:STATe?
<x> = 1 to 4, <y> = 1 to 4
Example :CURSOR:FFT:MARKER:M1:DY2:STATE ON
:CURSOR:FFT:MARKER:M1:DY2:STATE? ->
:CURSOR:FFT:MARKER:M1:DY2:STATE 1

:CURSor:FFT:MARKer:M<x>:DY<y>:VALue?
Function Queries the ΔY value between the FFT cursors (marker cursors).
Syntax :CURSor:FFT:MARKer:M<x>:DF<y>:VALue?
<x> = 1 to 4, <y> = 1 to 4
Example :CURSOR:FFT:MARKER:M1:DY2:VALUE? ->
:CURSOR:FFT:MARKER:M1:DY2:
VALUE 3.9750000E+00

:CURSor:FFT:MARKer:M<x>:POSition
Function Sets or queries the position of an FFT cursor (marker cursor).
Syntax :CURSor:FFT:MARKer:M<x>:
POSition {<NRf>}
:CURSor:FFT:MARKer:M<x>:POSition?
<x> = 1 to 4
<NRf> = –5 to 5
Example :CURSOR:FFT:MARKER:M1:POSITION -3.00
:CURSOR:FFT:MARKER:M1:POSITION? ->
:CURSOR:FFT:MARKER:M1:POSITION -3.00

:CURSor:FFT:MARKer:M<x>:TRACe

Function Sets or queries the source waveform that you want to measure using the FFT cursor (marker cursor).

Syntax
```
:CURSor:FFT:MARKer:M<x>:TRACe {OFF|FFT1|FFT2}
:CURSor:FFT:MARKer:M<x>:TRACe?
<x> = 1 to 4
```

Example
```
:CURSOR:FFT:MARKER:M1:TRACE FFT1
:CURSOR:FFT:MARKER:M1:TRACE? ->
:CURSOR:FFT:MARKER:M1:TRACE FFT1
```

:CURSor:FFT:MARKer:M<x>:F?

Function Queries all frequency-axis settings of an FFT cursor (marker cursor).

Syntax
```
:CURSor:FFT:MARKer:M<x>:F?
<x> = 1 to 4
```

:CURSor:FFT:MARKer:M<x>:F:STATe

Function Sets or queries whether the frequency-axis value of an FFT cursor (marker cursor) is displayed.

Syntax
```
:CURSor:FFT:MARKer:M<x>:F:STATe {<Boolean>}
:CURSor:FFT:MARKer:M<x>:F:STATe?
<x> = 1 to 4
```

Example
```
:CURSOR:FFT:MARKER:M1:F:STATE ON
:CURSOR:FFT:MARKER:M1:F:STATE? ->
:CURSOR:FFT:MARKER:M1:F:STATE 1
```

:CURSor:FFT:MARKer:M<x>:F:VALue?

Function Queries the frequency-axis value of an FFT cursor (marker cursor).

Syntax
```
:CURSor:FFT:MARKer:M<x>:F:VALue?
<x> = 1 to 4
```

Example
```
:CURSOR:FFT:MARKER:M1:F:VALUE? ->
:CURSOR:FFT:MARKER:M1:F:
VALUE 100.00000E+03
```

:CURSor:FFT:MARKer:M<x>:Y?

Function Queries all Y-axis settings of an FFT cursor (marker cursor).

Syntax
```
:CURSor:FFT:MARKer:M<x>:Y?
<x> = 1 to 4
```

:CURSor:FFT:MARKer:M<x>:Y:STATe

Function Sets or queries whether the Y-axis value of an FFT cursor (marker cursor) is displayed.

Syntax
```
:CURSor:FFT:MARKer:M<x>:Y:STATe {<Boolean>}
:CURSor:FFT:MARKer:M<x>:Y:STATe?
<x> = 1 to 4
```

Example
```
:CURSOR:FFT:MARKER:M1:Y:STATE ON
:CURSOR:FFT:MARKER:M1:Y:STATE? ->
:CURSOR:FFT:MARKER:M1:Y:STATE 1
```

:CURSor:FFT:MARKer:M<x>:Y:VALue?

Function Queries the Y-axis value of an FFT cursor (marker cursor).

Syntax
```
:CURSor:FFT:MARKer:M<x>:Y:VALue?
<x> = 1 to 4
```

Example
```
:CURSOR:FFT:MARKER:M1:Y:VALUE? ->
:CURSOR:FFT:MARKER:M1:Y:
VALUE -46.750000E+00
```

:CURSor:FFT:MARKer:FORM

Function Sets or queries the form of an FFT cursor (marker cursor).

Syntax
```
:CURSor:FFT:MARKer:FORM {LINE|MARK}
:CURSor:FFT:MARKer:FORM?
```

Example
```
:CURSOR:FFT:MARKER:FORM LINE
:CURSOR:FFT:MARKER:FORM? -> :CURSOR
:FFT:MARKER:FORM LINE
```

:CURSor:FFT:PEAK<x>?

Function Queries all settings of an FFT cursor (peak cursor).

Syntax
```
:CURSor:FFT:PEAK<x>?
<x> = 1, 2 (cursors 1 and 2)
Cursor 1 is dedicated for use as FFT1. Cursor 2 is dedicated for use as FFT2.
```

:CURSor:FFT:PEAK<x>:F?

Function Queries all frequency-axis settings of an FFT cursor (peak cursor).

Syntax
```
:CURSor:FFT:PEAK<x>:F?
<x> = 1, 2 (cursors 1 and 2)
```

:CURSor:FFT:PEAK<x>:F:STATe

Function Sets or queries whether the frequency-axis value of an FFT cursor (peak cursor) is displayed.

Syntax
```
:CURSor:FFT:PEAK<x>:F:STATe {<Boolean>}
:CURSor:FFT:PEAK<x>:F:STATe?
<x> = 1, 2 (cursors 1 and 2)
```

Example
```
:CURSOR:FFT:PEAK1:F:STATE ON
:CURSOR:FFT:PEAK1:F:STATE? ->
:CURSOR:FFT:PEAK1:F:STATE 1
```

:CURSor:FFT:PEAK<x>:F:VALue?

Function Queries the frequency-axis value of an FFT cursor (peak cursor).

Syntax
```
:CURSor:FFT:PEAK<x>:F:VALue?
<x> = 1, 2 (cursors 1 and 2)
```

Example
```
:CURSOR:FFT:PEAK1:F:VALUE? ->
:CURSOR:FFT:PEAK1:F:
VALUE 2.0000000E+03
```

:CURSor:FFT:PEAK<x>:Y?
Function Queries all Y-axis settings of an FFT cursor (peak cursor).
Syntax
```
:CURSor:FFT:PEAK<x>:Y?
<x> = 1, 2 (cursors 1 and 2)
```

:CURSor:FFT:PEAK<x>:Y:STATe
Function Sets or queries whether the Y-axis value of an FFT cursor (peak cursor) is displayed.
Syntax
```
:CURSor:FFT:PEAK<x>:Y:
STATe {<Boolean>}
:CURSor:FFT:PEAK<x>:Y:STATe?
<x> = 1, 2 (cursors 1 and 2)
```
Example
```
:CURSOR:FFT:PEAK2:Y:STATE ON
:CURSOR:FFT:PEAK2:Y:STATE? ->
:CURSOR:FFT:PEAK2:Y:STATE 1
```

:CURSor:FFT:PEAK<x>:Y:VALue?
Function Queries the Y-axis value of an FFT cursor (peak cursor).
Syntax
```
:CURSor:FFT:PEAK<x>:Y:VALue?
<x> = 1, 2 (cursors 1 and 2)
```
Example
```
:CURSOR:FFT:PEAK2:Y:VALUE? ->
:CURSOR:FFT:PEAK2:Y:
VALUE 30.933333E+00
```

:CURSor:FFT:TYPE
Function Sets or queries the FFT cursor type.
Syntax
```
:CURSor:FFT:TYPE {OFF|PEAK|MARKer}
:CURSor:FFT:TYPE?
```
Example
```
:CURSOR:FFT:TYPE PEAK
:CURSOR:FFT:TYPE? -> :CURSOR:FFT:
TYPE PEAK
```

:CURSor[:TY]?
Function Queries all cursor settings for the T-Y display.
Syntax
```
:CURSor[:TY]?
```

:CURSor[:TY]:DEGRee?
Function Queries all angle cursor settings for the T-Y display.
Syntax
```
:CURSor[:TY]:DEGRee?
```

:CURSor[:TY]:DEGRee:D<X>?
Function Queries all angle difference (D<x>) settings for the angle cursors on the T-Y display.
Syntax
```
:CURSor:TY:DEGRee:D<X>?
<x> = 1, 2
```
Example
```
:CURSOR:TY:DEGREE:D1:STATE? ->
:CURSOR:TY:DEGREE:D1:STATE 1
```

:CURSor[:TY]:DEGRee:D<X>:STATe
Function Sets or queries whether the angle difference (D<x>) between the angle cursors on the T-Y display is displayed.
Syntax
```
:CURSor:TY:DEGRee:D<X>:
STATe {<Boolean>}
:CURSor:TY:DEGRee:D<X>:STATe?
<x> = 1, 2
```
Example
```
:CURSOR:TY:DEGREE:D1:STATE ON
:CURSOR:TY:DEGREE:D1:STATE? ->
:CURSOR:TY:DEGREE:D1:STATE 1
```

:CURSor[:TY]:DEGRee:D<X>:VALue?
Function Queries the angle difference (D<x>) between the angle cursors on the T-Y display.
Syntax
```
:CURSor:TY:DEGRee:D<X>:VALue?
<x> = 1, 2
```
Example
```
:CURSOR:TY:DEGREE:D1:VALUE? ->
:CURSOR:TY:DEGREE:D1:
VALUE -179.55000E+00
```

:CURSor[:TY]:DEGRee:DD?
Function Queries all angle difference (Δd value) settings. The angle difference is between the angle cursors on the T-Y display.
Syntax
```
:CURSor:TY:DEGRee:DD?
```

:CURSor[:TY]:DEGRee:DD:STATe
Function Sets or queries whether the angle difference (Δd) value between the angle cursors on the T-Y display is displayed.
Syntax
```
:CURSor:TY:DEGRee:DD:
STATe {<Boolean>}
:CURSor:TY:DEGRee:DD:STATe?
```
Example
```
:CURSOR:TY:DEGREE:DD:STATE ON
:CURSOR:TY:DEGREE:DD:STATE? ->
:CURSOR:TY:DEGREE:DD:STATE 1
```

:CURSor[:TY]:DEGRee:DD:VALue?
Function Queries the angle difference (Δd value) between the angle cursors on the T-Y display.
Syntax
```
:CURSor:TY:DEGRee:DD:VALue?
```
Example
```
:CURSOR:TY:DEGREE:DD:VALUE? ->
:CURSOR:TY:DEGREE:DD:
VALUE 719.55000E+00
```

:CURSor[:TY]:DEGRee:DY?
Function Queries all angle difference (ΔY value) settings. The angle difference is between the angle cursors on the T-Y display.
Syntax
```
:CURSor:TY:DEGRee:DY?
```

:CURSor[:TY]:DEGRee:DY:STATe

Function Sets or queries whether the angle difference (ΔY) value between the angle cursors on the T-Y display is displayed.

Syntax
```
:CURSor:TY:DEGRee:DY:
STATe {<Boolean>}
:CURSor:TY:DEGRee:DY:STATe?
```

Example
```
:CURSOR:TY:DEGREE:DY:STATE ON
:CURSOR:TY:DEGREE:DY:STATE? ->
:CURSOR:TY:DEGREE:DY:STATE 1
```

:CURSor[:TY]:DEGRee:DY:VALue?

Function Queries the angle difference (ΔY value) between the angle cursors on the T-Y display.

Syntax
```
:CURSor:TY:DEGRee:DY:VALue?
```

Example
```
:CURSOR:TY:DEGREE:DY:
VALUE 0.0000000E+00
```

:CURSor[:TY]:DEGRee:POSition<x>

Function Sets or queries an angle cursor position on the T-Y display.

Syntax
```
:CURSor:TY:DEGRee:
POSition<x> {<NRf>}
:CURSor:TY:DEGRee:POSition<x>?
<x> = 1, 2
<NRf> = -5 to 5 (the resolution
depends on the measurement length)
```

Example
```
:CURSOR:TY:DEGREE:POSITION1 1
:CURSOR:TY:DEGREE:POSITION1? ->
:CURSOR:TY:DEGREE:
POSITION1 1.000000000000
```

:CURSor[:TY]:DEGRee:REFerence<x>

Function Sets or queries the angle reference start (Ref1) or end (Ref2) point on the T-Y display.

Syntax
```
:CURSor:TY:DEGRee:
REFerence<x> {<NRf>}
:CURSor:TY:DEGRee:REFerence<x>?
<x> = 1, 2
<NRf> = -5 to 5 (the resolution
depends on the measurement length)
```

Example
```
:CURSOR:TY:DEGREE:REFERENCE1 -1
:CURSOR:TY:DEGREE:REFERENCE1? ->
:CURSOR:TY:DEGREE:
REFERENCE1 -1.000000000000
```

:CURSor[:TY]:DEGRee:RVALue

Function Sets or queries an angle cursor's reference angle on the T-Y display.

Syntax
```
:CURSor:TY:DEGRee:RVALue {<NRf>}
:CURSor:TY:DEGRee:RVALue?
<NRf> = 1 to 720 (in steps of 1)
```

Example
```
:CURSOR:TY:DEGREE:RVALUE 360
:CURSOR:TY:DEGREE:RVALUE? ->
:CURSOR:TY:DEGREE:RVALUE 360
```

:CURSor[:TY]:DEGRee:TRACe

Function Sets or queries the source waveform that you want to measure using the angle cursors on the T-Y display.

Syntax
```
:CURSor:TY:DEGRee:
TRACe {<NRf>[,<NRf>]|MATH<x>|ALL}
:CURSor:TY:DEGRee:TRACe?
<NRf> = 1 to 16
MATH<x>'s <x> = 1 to 8
<NRf> = 1 to 16 (only when it is
necessary to specify the sub channel)
```

Example
```
:CURSOR:TY:DEGREE:TRACE 1
:CURSOR:TY:DEGREE:TRACE? -> :CURSOR
:TY:DEGREE:TRACE 1
```

:CURSor[:TY]:DEGRee:Y<x>?

Function Queries all measured-value settings for an angle cursor on the T-Y display.

Syntax
```
:CURSor:TY:DEGRee:Y<x>?
<x> = 1, 2
```

:CURSor[:TY]:DEGRee:Y<x>:STATe

Function Sets or queries whether the measured value of an angle cursor on the T-Y display is displayed.

Syntax
```
:CURSor:TY:DEGRee:Y<x>:
STATe {<Boolean>}
:CURSor:TY:DEGRee:Y<x>:STATe?
<x> = 1, 2
```

Example
```
:CURSOR:TY:DEGREE:Y1:STATE 1
:CURSOR:TY:DEGREE:Y1:STATE? ->
:CURSOR:TY:DEGREE:Y1:STATE 1
```

:CURSor[:TY]:DEGRee:Y<x>:VALue?

Function Queries the measured value of an angle cursor on the T-Y display.

Syntax
```
:CURSor:TY:DEGRee:Y<x>:VALue?
<x> = 1, 2
```

Example
```
:CURSOR:TY:DEGREE:Y1:VALUE? ->
:CURSOR:TY:DEGREE:Y1:
VALUE 0.0000000E+00
```

Description
- If you have set the waveform that you want to measure to ALL, all the measured values from the channels, sub channels, and Math channels of all the installed modules will be returned.
- "NAN" will be returned for any unused channels and channels that do not have modules installed in their corresponding slots.
- For modules that have sub channels, the measured values of all sub channels will be returned. "NAN" will be returned for any sub channels whose input is turned off.

:CURSor[:TY]:HORizontal?

Function Queries all horizontal cursor settings for the T-Y display.

Syntax
```
:CURSor:TY:HORizontal?
```

:CURSor[:TY]:HORizontal:DY?

Function Queries all ΔY-axis settings of the horizontal cursors on the T-Y display.

Syntax
```
:CURSor:TY:HORizontal:DY?
```

:CURSor[:TY]:HORizontal:DY:STATe

Function Sets or queries whether the ΔY-axis value between the horizontal cursors on the T-Y display is displayed.

Syntax
```
:CURSor:TY:HORizontal:DY:
STATe {<Boolean>}
:CURSor:TY:HORizontal:DY:STATe?
```

Example
```
:CURSOR:TY:HORIZONTAL:DY:STATE ON
:CURSOR:TY:HORIZONTAL:DY:STATE? ->
:CURSOR:TY:HORIZONTAL:DY:STATE 1
```

:CURSor[:TY]:HORizontal:DY:VALue?

Function Queries the ΔY-axis value of the horizontal cursors on the T-Y display.

Syntax
```
:CURSor:TY:HORizontal:DY:VALue?
```

Example
```
:CURSOR:TY:HORIZONTAL:DY:VALUE? ->
:CURSOR:TY:HORIZONTAL:DY:
VALUE 300.00000E+00
```

:CURSor[:TY]:HORizontal:POSition<x>

Function Sets or queries a horizontal cursor position on the T-Y display.

Syntax
```
:CURSor:TY:HORizontal:
POSition<x> {<NRf>}
:CURSor:TY:HORizontal:POSition<x>?
<x> = 1, 2
<NRf> = -5 to 5 (in steps of 1/100)
```

Example
```
:CURSOR:TY:HORIZONTAL:POSITION2 -3
:CURSOR:TY:HORIZONTAL:POSITION2? ->
:CURSOR:TY:HORIZONTAL:
POSITION2 -3.00
```

:CURSor[:TY]:HORizontal:TRACe

Function Sets or queries the source waveform that you want to measure using the horizontal cursors on the T-Y display.

Syntax
```
:CURSor:TY:HORizontal: {<NRf>|
MATH<x>,<NRf>}
:CURSor:TY:HORizontal:TRACe?
<NRf> = 1 to 16
<x> = 1 to 8
<NRf> = 1 to 16 (only when it is
necessary to specify the sub channel)
```

Example
```
:CURSOR:TY:HORIZONTAL:TRACE 1
:CURSOR:TY:HORIZONTAL:TRACE? ->
:CURSOR:TY:HORIZONTAL:TRACE 1
```

:CURSor[:TY]:HORizontal:Y<x>?

Function Queries all Y-axis settings of the horizontal cursor on the T-Y display.

Syntax
```
:CURSor:TY:HORizontal:Y<x>?
<x> = 1, 2
```

:CURSor[:TY]:HORizontal:Y<x>:STATe

Function Sets or queries whether the Y-axis value for a horizontal cursor on the T-Y display is displayed.

Syntax
```
:CURSor:TY:HORizontal:Y<x>:
STATe {<Boolean>}
:CURSor:TY:HORizontal:Y<x>:STATe?
<x> = 1, 2
```

Example
```
:CURSOR:TY:HORIZONTAL:Y1:STATE ON
:CURSOR:TY:HORIZONTAL:Y1:STATE? ->
:CURSOR:TY:HORIZONTAL:Y1:STATE 1
```

:CURSor[:TY]:HORizontal:Y<x>:VALue?

Function Queries the Y-axis value of a horizontal cursor on the T-Y display.

Syntax
```
:CURSor:TY:HORizontal:Y<x>:VALue?
<x> = 1, 2
```

Example
```
:CURSOR:TY:HORIZONTAL:Y1:VALUE? ->
:CURSOR:TY:HORIZONTAL:Y1:
VALUE 150.00000E+00
```

:CURSor[:TY]:MARKer?

Function Queries all marker cursor settings for the T-Y display.

Syntax
```
:CURSor[:TY]:MARKer?
```

:CURSor[:TY]:MARKer:M<x>?

Function Queries all settings related to the marker cursor measurement items for the T-Y display.

Syntax
```
:CURSor[:TY]:MARKer:M<x>?
<x> = 1 to 4
```

:CURSor[:TY]:MARKer:M<x>:DX<y>?

Function Queries all ΔX value settings. ΔX is between the marker cursors on the T-Y display.

Syntax
```
:CURSor:TY:MARKer:M<x>:DX<y>?
<x> = 1 to 4
<y> = 1 to 4
```

:CURSor[:TY]:MARKer:M<x>:DX<y>:STATe

Function Sets or queries whether the ΔX value between the marker cursors on the T-Y display is displayed.

Syntax
```
:CURSor:TY:MARKer:M<x>:DX<y>?
<x> = 1 to 4
<y> = 1 to 4
```

Example
```
:CURSOR:TY:MARKER:M1:DX2:STATE ON
:CURSOR:TY:MARKER:M1:DX2:STATE? ->
:CURSOR:TY:MARKER:M1:DX2:STATE 1
```

:CURSor[:TY]:MARKer:M<x>:DX<y>:VALue?

Function Queries the ΔX value between the marker cursors on the T-Y display.

Syntax
```
:CURSor:TY:MARKer:M<x>:DX<y>:VALue?
<x> = 1 to 4
<y> = 1 to 4
```

Example
```
:CURSOR:TY:MARKER:M1:DX2:VALUE? ->
:CURSOR:TY:MARKER:M1:DX2:
VALUE 2.0000000E-03
```

:CURSor[:TY]:MARKer:M<x>:DY<y>?

Function Queries all ΔY value settings. ΔY is between the marker cursors on the T-Y display.

Syntax
```
:CURSor:TY:MARKer:M<x>:DY<y>?
<x> = 1 to 4
<y> = 1 to 4
```

:CURSor[:TY]:MARKer:M<x>:DY<y>:STATe

Function Sets or queries whether the ΔY value between the marker cursors on the T-Y display is displayed.

Syntax
```
:CURSor:TY:MARKer:M<x>:DY<y>?
<x> = 1 to 4
<y> = 1 to 4
```

Example
```
:CURSOR:TY:MARKER:M1:DY2:STATE ON
:CURSOR:TY:MARKER:M1:DY2:STATE? ->
:CURSOR:TY:MARKER:M1:DY2:STATE 1
```

:CURSor[:TY]:MARKer:M<x>:DY<y>:VALue?

Function Queries the ΔY value between the marker cursors on the T-Y display.

Syntax
```
:CURSor:TY:MARKer:M<x>:DY<y>:VALue?
<x> = 1 to 4
<y> = 1 to 4
```

Example
```
:CURSOR:TY:MARKER:M1:DY2:VALUE? ->
:CURSOR:TY:MARKER:M1:DY2:
VALUE 0.0000000E+00
```

:CURSor[:TY]:MARKer:M<x>:POSition

Function Sets or queries a marker cursor position on the T-Y display.

Syntax
```
:CURSor:TY:MARKer:M<x>:
POSition {<NRf>}
:CURSor:TY:MARKer:M<x>:POSition?
<x> = 1 to 4
<NRf> = -5 to 5
```

Example
```
:CURSOR:TY:MARKER:M1:POSITION -3
:CURSOR:TY:MARKER:M1:POSITION? ->
:CURSOR:TY:MARKER:M1:
POSITION -3.000000000000
```

:CURSor[:TY]:MARKer:M<x>:TRACe

Function Sets or queries the source waveform that you want to measure using the marker cursors on the T-Y display.

Syntax
```
:CURSor:TY:MARKer:M<x>:TRACe {OFF|
<NRf>|MATH<x>,<NRf>}
:CURSor:TY:MARKer:M<x>:TRACe?
<NRf> = 1 to 16
<x> = 1 to 8
<NRf> = 1 to 16 (only when it is
necessary to specify the sub channel)
```

Example
```
:CURSOR:TY:MARKER:M1:TRACE 1
:CURSOR:TY:MARKER:M1:TRACE? ->
:CURSOR:TY:MARKER:M1:TRACE 1
```

:CURSor[:TY]:MARKer:M<x>:X?

Function Queries all X-axis settings for a marker cursor on the T-Y display.

Syntax
```
:CURSor:TY:MARKer:M<x>:X?
<x> = 1 to 4
```

:CURSor[:TY]:MARKer:M<x>:X:STATe

Function Sets or queries whether the X-axis value for a marker cursor on the T-Y display is displayed.

Syntax
```
:CURSor:TY:MARKer:M<x>:X:
STATe {<Boolean>}
:CURSor:TY:MARKer:M<x>:X:STATe?
<x> = 1 to 4
```

Example
```
:CURSOR:TY:MARKER:M4:X:STATE ON
:CURSOR:TY:MARKER:M4:X:STATE? ->
:CURSOR:TY:MARKER:M4:X:STATE 1
```

:CURSor[:TY]:MARKer:M<x>:X:VALue?

Function Queries the X-axis value of a marker cursor on the T-Y display.

Syntax
```
:CURSor:TY:MARKer:M<x>:X:STATe?
<x> = 1 to 4
```

Example
```
:CURSOR:TY:MARKER:M4:X:VALUE? ->
:CURSOR:TY:MARKER:M4:X:
VALUE 3.0000000E-03
```

:CURSor[:TY]:MARKer:M<x>:Y?

Function Queries all Y-axis settings for a marker cursor on the T-Y display.

Syntax
```
:CURSor:TY:MARKer:M<x>:Y?
<x> = 1 to 4
```

:CURSor[:TY]:MARKer:M<x>:Y:STATe

Function Sets or queries whether the Y-axis value for a marker cursor on the T-Y display is displayed.

Syntax
```
:CURSor:TY:MARKer:M<x>:Y:
STATe {<Boolean>}
:CURSor:TY:MARKer:M<x>:Y:STATe?
<x> = 1 to 4
```

Example
```
:CURSOR:TY:MARKER:M4:Y:STATE ON
:CURSOR:TY:MARKER:M4:Y:STATE? ->
:CURSOR:TY:MARKER:M4:Y:STATE 1
```

:CURSor[:TY]:MARKer:M<x>:Y:VALue?

Function Queries the Y-axis value of a marker cursor on the T-Y display.

Syntax
```
:CURSor:TY:MARKer:M<x>:Y:STATe?
<x> = 1 to 4
```

Example
```
:CURSOR:TY:MARKER:M4:Y:VALUE? ->
:CURSOR:TY:MARKER:M4:Y:
VALUE 41.666667E-03
```

:CURSor[:TY]:MARKer:FORM

Function Sets or queries the form of the marker cursors on the T-Y display.

Syntax
```
:CURSor[:TY]:MARKer:FORM {LINE|MARK}
:CURSor[:TY]:MARKer:FORM?
```

Example
```
:CURSOR:TY:MARKER:FORM MARK
:CURSOR:TY:MARKER:FORM? -> :CURSOR:
TY:MARKER:FORM MARK
```

:CURSor[:TY]:TYPE

Function Sets or queries the cursor type on the T-Y display.

Syntax
```
:CURSor:TY:TYPE {OFF|HORizontal|
VERTical|MARKer|DEGRee|HAVertical}
:CURSor:TY:TYPE?
```

Example
```
:CURSOR:TY:TYPE MARKER
:CURSOR:TY:TYPE? -> :CURSOR:TY:
TYPE MARKER
```

:CURSor[:TY]:VERTical?

Function Queries all V cursor settings for the T-Y display.

Syntax
```
:CURSor[:TY]:VERTical?
```

:CURSor[:TY]:VERTical:DX?

Function Queries all ΔX value settings. ΔX is between the V cursors on the T-Y display.

Syntax
```
:CURSor:TY:VERTical:DX?
```

:CURSor[:TY]:VERTical:DX:STATe

Function Sets or queries whether the ΔX value between the V cursors on the T-Y display is displayed.

Syntax
```
:CURSor:TY:VERTical:DX:
STATe {<Boolean>}
:CURSor:TY:VERTical:DX:STATe?
```

Example
```
:CURSOR:TY:VERTICAL:DX:STATE 1
:CURSOR:TY:VERTICAL:DX:STATE? ->
:CURSOR:TY:VERTICAL:DX:STATE 1
```

:CURSor[:TY]:VERTical:DX:VALue?

Function Queries the ΔX value between the V cursors on the T-Y display

Syntax
```
:CURSor:TY:VERTical:DX:VALue?
```

Example
```
:CURSOR:TY:VERTICAL:DX:VALUE? ->
:CURSOR:TY:VERTICAL:DX:
VALUE 3.0000000E-03
```

Description
- When the Internal Clock Is Being Used as the Time Base
  When measurements are performed in the time domain, this queries the time between markers.
  When measurements are performed in the frequency domain, this queries the frequency between markers.
- When an External Clock is Being Used as the Time Base
  This queries the number of points between marker cursors.

:CURSor[:TY]:VERTical:DY?

Function Queries all Y-axis-value settings for V cursors on the T-Y display.

Syntax
```
:CURSor[:TY]:VERTical:DY?
```

:CURSor[:TY]:VERTical:DY:STATe

Function Sets or queries whether the Y-axis value for V cursors on the T-Y display is displayed.

Syntax
```
:CURSor:TY:VERTical:DY:
STATe {<Boolean>}
:CURSor:TY:VERTical:DY:STATe?
```

Example
```
:CURSOR:TY:VERTICAL:DY:STATE ON
:CURSOR:TY:VERTICAL:DY:STATE? ->
:CURSOR:TY:VERTICAL:DY:STATE 1
```

:CURSor[:TY]:VERTical:DY:VALue?

Function Queries the Y-axis value of V cursors on the T-Y display.

Syntax
```
:CURSor:TY:VERTical:DY:VALue?
```

:CURSor[:TY]:VERTical:PERDt?

Function Queries all 1/ΔT value settings. 1/ΔT is between the V cursors on the T-Y display.

Syntax
```
:CURSor:TY:VERTical:PERDt?
```

:CURSor[:TY]:VERTical:PERDt:STATe

Function Sets or queries whether the 1/ΔT value between the V cursors on the T-Y display is displayed.

Syntax
```
:CURSor:TY:VERTical:PERDt:
STATe {<Boolean>}
:CURSor:TY:VERTical:PERDt:STATe?
```

Example
```
:CURSOR:TY:VERTICAL:PERDT:STATE ON
:CURSOR:TY:VERTICAL:PERDT:STATE? ->
:CURSOR:TY:VERTICAL:PERDT:STATE 1
```

:CURSor[:TY]:VERTical:PERDt:VALue?

Function Queries the 1/ΔT value between the V cursors on the T-Y display

Syntax
```
:CURSor:TY:VERTical:PERDt:VALue?
```

Example
```
:CURSOR:TY:VERTICAL:PERDT:
VALUE 250.00000E+00
```

Description If the source waveform is being measured in the frequency domain, “NAN” is returned.

:CURSor[:TY]:VERTical:POSition<x>

Function Sets or queries a V cursor position on the T-Y display.

Syntax
```
:CURSor:TY:VERTical:
POSition<x> {<NRf>}
:CURSor:TY:VERTical:POSition<x>?
<x> = 1, 2
<NRf> = -5 to 5 (the resolution
depends on the measurement length)
```

Example
```
:CURSOR:TY:VERTICAL:POSITION2 4
:CURSOR:TY:VERTICAL:POSITION2? ->
:CURSOR:TY:VERTICAL:
POSITION2 4.000000000000
```

:CURSor[:TY]:VERTical:TRACe

Function Sets or queries the source waveform that you want to measure using the V cursors on the T-Y display.

Syntax
```
:CURSor:TY:VERTical:
TRACe {<NRf>[,<NRf>]|MATH<x>|ALL}
:CURSor:TY:VERTical:TRACe?
<NRf> = 1 to 16
MATH<x>'s <x> = 1 to 8
<NRf> = 1 to 16 (only when it is
necessary to specify the sub channel)
```

Example
```
:CURSOR:TY:VERTICAL:TRACE 2
:CURSOR:TY:VERTICAL:TRACE? ->
:CURSOR:TY:VERTICAL:TRACE 2
```

:CURSor[:TY]:VERTical:X<x>?

Function Queries all X-axis-value settings for a V cursor on the T-Y display.

Syntax
```
:CURSor:TY:VERTical:X<x>?
<x> = 1, 2
```

:CURSor[:TY]:VERTical:X<x>:STATe

Function Sets or queries whether the X-axis value for a V cursor on the T-Y display is displayed.

Syntax
```
:CURSor:TY:VERTical:X<x>:
STATe {<Boolean>}
:CURSor:TY:VERTical:X<x>:STATe?
<x> = 1, 2
```

Example
```
:CURSOR:TY:VERTICAL:X1:STATE 1
:CURSOR:TY:VERTICAL:X1:STATE? ->
:CURSOR:TY:VERTICAL:X1:STATE 1
```

:CURSor[:TY]:VERTical:X<x>:VALue?

Function Queries the X-axis value of a V cursor on the T-Y display.

Syntax
```
:CURSor:TY:VERTical:X<x>:VALue?
<x> = 1, 2
```

Example
```
:CURSOR:TY:VERTICAL:X1:VALUE? ->
:CURSOR:TY:VERTICAL:X1:
VALUE 0.0000000E+00
```

Description
- When the Internal Clock Is Being Used as the Time Base
  When measurements are performed in the time domain, this queries the time. When measurements are performed in the frequency domain, this queries the frequency.
- When an External Clock Is Being Used as the Time Base
  When measurements are performed in the time domain, this queries the number of points using the trigger point as the reference. When measurements are performed in the frequency domain, this queries the number of points using the front of the waveform as the reference.

:CURSor[:TY]:VERTical:Y<x>?

Function Queries all Y-axis-value settings for a V cursor on the T-Y display.

Syntax
```
:CURSor:TY:VERTical:X<x>?
<x> = 1, 2
```

:CURSor[:TY]:VERTical:Y<x>:STATe

Function Sets or queries whether the Y-axis value for a V cursor on the T-Y display is displayed.

Syntax
```
:CURSor:TY:VERTical:Y<x>:
STATe {<Boolean>}
:CURSor:TY:VERTical:Y<x>?
<x> = 1, 2
```

Example
```
:CURSOR:TY:VERTICAL:Y1:STATE 1
:CURSOR:TY:VERTICAL:Y1:STATE? ->
:CURSOR:TY:VERTICAL:Y1:STATE 1
```

:CURSor[:TY]:VERTical:Y<x>:VALue?

Function Queries the Y-axis value (measured value) of a V cursor on the T-Y display.

Syntax
```
:CURSor:TY:VERTical:Y<x>:VALue?
<x> = 1, 2
```

Example
```
:CURSOR:TY:VERTICAL:Y2:VALUE? ->
:CURSOR:TY:VERTICAL:Y2:
VALUE -333.33333E-03
```

Description
- When the Internal Clock Is Being Used as the Time Base
  When measurements are performed in the time domain, this queries the time. When measurements are performed in the frequency domain, this queries the frequency.
- If you have set the waveform that you want to measure to ALL, all the measured values from the channels, sub channels, and Math channels of all the installed modules will be returned.
- "NAN" will be returned for any unused channels and channels that do not have modules installed in their corresponding slots.
- For modules that have sub channels, the measured values of all sub channels will be returned. "NAN" will be returned for any sub channels whose input is turned off.

:CURSor:XY?

Function Queries all cursor settings for the X-Y display.

Syntax
```
:CURSor:XY?
```

:CURSor:XY:HORizontal?

Function Queries all H cursor settings for the X-Y display.

Syntax
```
:CURSor:XY:HORizontal?
```

:CURSor:XY:HORizontal:DY?

Function Queries all ΔY-axis-value settings. The ΔY-axis value is between the H cursors on the X-Y display.

Syntax
```
:CURSor:XY:HORizontal:DY?
```

:CURSor:XY:HORizontal:DY:STATe

Function Sets or queries whether the ΔY-axis value between the H cursors on the X-Y display is displayed.

Syntax
```
:CURSor:XY:HORizontal:DY:
STATe {<Boolean>}
:CURSor:XY:HORizontal:DY:STATe?
```

Example
```
:CURSOR:XY:HORIZONTAL:DY:STATE ON
:CURSOR:XY:HORIZONTAL:DY:STATE? ->
:CURSOR:XY:HORIZONTAL:DY:STATE 1
```

:CURSor:XY:HORizontal:DY:VALue?

Function Queries the ΔY-axis value between the H cursors on the X-Y display

Syntax
```
:CURSor:XY:HORizontal:DY:VALue?
```

Example
```
:CURSOR:XY:HORIZONTAL:DY:VALUE? ->
:CURSOR:XY:HORIZONTAL:DY:
VALUE 300.00000E+00
```

:CURSor:XY:HORizontal:POSition<x>

Function Sets or queries H cursor positions on the X-Y display.

Syntax
```
:CURSor:XY:HORizontal:
POSition<x> {<NRf>}
:CURSor:XY:HORizontal:POSition<x>?
<x> = 1, 2
<NRf> = -5 to 5 (in steps of 1/100)
```

Example
```
:CURSOR:XY:HORIZONTAL:POSITION1 3
:CURSOR:XY:HORIZONTAL:POSITION1? ->
:CURSOR:XY:HORIZONTAL:POSITION1 3.00
```

Description You can query cursor position information for cursors XY1 to XY8. Use the :CURSor:XY:HORizontal:TRACe command to specify the waveform that you want to query.

:CURSor:XY:HORizontal:TRACe

Function Sets or queries the source waveform that you want to measure using the H cursors on the X-Y display.

Syntax
```
:CURSor:XY:HORizontal:TRACe {XY1|
XY2|XY3|XY4|XY5|XY6|XY7|XY8}
:CURSor:XY:HORizontal:TRACe?
```

Example
```
:CURSOR:XY:HORIZONTAL:TRACE XY2
:CURSOR:XY:HORIZONTAL:TRACE? ->
:CURSOR:XY:HORIZONTAL:TRACE XY2
```

:CURSor:XY:HORizontal:Y<x>?

Function Queries all Y-axis-value settings for an H cursor on the X-Y display.

Syntax
```
:CURSor:XY:HORizontal:Y<x>?
```

:CURSor:XY:HORizontal:Y<x>:STATe

Function Sets or queries whether the Y-axis value for an H cursor on the X-Y display is displayed.

Syntax
```
:CURSor:XY:HORizontal:Y<x>:
STATe {<Boolean>}
:CURSor:XY:HORizontal:Y<x>:STATe?
<x> = 1, 2
```

Example
```
:CURSOR:XY:HORIZONTAL:Y1:STATE ON
:CURSOR:XY:HORIZONTAL:Y1:STATE? ->
:CURSOR:XY:HORIZONTAL:Y1:STATE 1
```

:CURSor:XY:HORizontal:Y<x>:VALue?

Function Queries the Y-axis value of an H cursor on the X-Y display

Syntax
```
:CURSor:XY:HORizontal:Y<x>:VALue?
<x> = 1, 2
```

Example
```
:CURSOR:XY:HORIZONTAL:Y1:VALUE? ->
:CURSOR:XY:HORIZONTAL:Y1:
VALUE 150.00000E+00
```

:CURSor:XY:MARKer:M<x>?

Function Queries all marker cursor settings for the X-Y display.

Syntax
```
:CURSor:XY:MARKer:M<x>?
<x> = 1 to 4
```

:CURSor:XY:MARKer:M<x>:POSition

Function Sets or queries a marker cursor's time-axis-equivalent position on the X-Y display.

Syntax
```
:CURSor:XY:MARKer:M<x>:
POSition {<NRf>}
:CURSor:XY:MARKer:M<x>:POSition?
<x> = 1 to 4
<NRf> = -5 to 5div (the resolution
depends on the measurement length)
```

Example
```
:CURSOR:XY:MARKER:M1:POSITION -3
:CURSOR:XY:MARKER:M1:POSITION? ->
:CURSOR:XY:MARKER:M1:
POSITION -3.000000000000
```

:CURSor:XY:MARKer:M<x>:T?

Function Queries all time value settings for a marker cursor on the X-Y display.

Syntax
```
:CURSor:XY:MARKer:M<x>:T?
<x> = 1 to 4
```

:CURSor:XY:MARKer:M<x>:T:STATe

Function Sets or queries whether the time value for a marker cursor on the X-Y display is displayed.

Syntax
```
:CURSor:XY:MARKer:M<x>:T:
STATe {<Boolean>}
:CURSor:XY:MARKer:M<x>:T:STATe?
<x> = 1 to 4
```

Example
```
:CURSOR:XY:MARKER:M1:T:STATE ON
:CURSOR:XY:MARKER:M1:T:STATE? ->
:CURSOR:XY:MARKER:M1:T:STATE 1
```

:CURSor:XY:MARKer:M<x>:T:VALue?

Function Queries the time value of a marker cursor on the X-Y display.

Syntax
```
:CURSor:XY:MARKer:M<x>:T:VALue?
<x> = 1 to 4
```

Example
```
:CURSOR:XY:MARKER:M1:T:VALUE? ->
:CURSOR:XY:MARKER:M1:T:
VALUE -3.0000000E-03
```

:CURSor:XY:MARKer:M<x>:TRACe

Function Sets or queries the source waveform that you want to measure using the marker cursors on the X-Y display.

Syntax
```
:CURSor:XY:MARKer:M<x>:TRACe {OFF|
XY1|XY2|XY3|XY4|XY5|XY6|XY7|XY8}
:CURSor:XY:MARKer:M<x>:TRACe?
<x> = 1 to 4
```

Example
```
:CURSOR:XY:MARKER:M1:TRACE XY1
:CURSOR:XY:MARKER:M1:TRACE? ->
:CURSOR:XY:MARKER:M1:TRACE XY1
```

:CURSor:XY:MARKer:M<x>:X?

Function Queries all X-axis-value settings for a marker cursor on the X-Y display.

Syntax
```
:CURSor:XY:MARKer:M<x>:X?
<x> = 1 to 4
```

:CURSor:XY:MARKer:M<x>:X:STATe

Function Sets or queries whether the X-axis value for a marker cursor on the X-Y display is displayed.

Syntax
```
:CURSor:XY:MARKer:M<x>:X:
STATe {<Boolean>}
:CURSor:XY:MARKer:M<x>:X:STATe?
<x> = 1 to 4
```

Example
```
:CURSOR:XY:MARKER:M1:X:STATE 1
:CURSOR:XY:MARKER:M1:X:STATE? ->
:CURSOR:XY:MARKER:M1:X:STATE 1
```

:CURSor:XY:MARKer:M<x>:X:VALue?

Function Queries the X-axis value of a marker cursor on the X-Y display.

Syntax
```
:CURSor:XY:MARKer:M<x>:X:VALue?
<x> = 1 to 4
```

Example
```
:CURSOR:XY:MARKER:M1:X:VALUE? ->
:CURSOR:XY:MARKER:M1:X:
VALUE 333.33333E-03
```

:CURSor:XY:MARKer:M<x>:Y?

Function Queries all Y-axis-value settings for a marker cursor on the X-Y display.

Syntax
```
:CURSor:XY:MARKer:M<x>:Y?
<x> = 1 to 4
```

:CURSor:XY:MARKer:M<x>:Y:STATe

Function Sets or queries whether the Y-axis value for a marker cursor on the X-Y display is displayed.

Syntax
```
:CURSor:XY:MARKer:M<x>:Y:
STATe {<Boolean>}
:CURSor:XY:MARKer:M<x>:Y:STATe?
<x> = 1 to 4
```

Example
```
:CURSOR:XY:MARKER:M1:Y:STATE ON
:CURSOR:XY:MARKER:M1:Y:STATE? ->
:CURSOR:XY:MARKER:M1:Y:STATE 1
```

:CURSor:XY:MARKer:M<x>:Y:VALue?

Function Queries the Y-axis value of a marker cursor on the X-Y display.

Syntax
```
:CURSor:XY:MARKer:M<x>:Y:VALue?
<x> = 1 to 4
```

Example
```
:CURSOR:XY:MARKER:M1:Y:VALUE? ->
:CURSOR:XY:MARKER:M1:Y:
VALUE 0.0000000E+00
```

:CURSor:XY:MARKer:FORM

Function Sets or queries the form of the marker cursors on the X-Y display.

Syntax
```
:CURSor:XY:MARKer:FORM {LINE|MARK}
:CURSor:XY:MARKer:FORM?
```

Example
```
:CURSOR:XY:MARKER:FORM LINE
:CURSOR:XY:MARKER:FORM? -> :CURSOR
:XY:MARKER:FORM LINE
```

Description `<x> = 1 to 4 (however, the setting is the same for cursors 1 to 4)`

:CURSor:XY:TYPE

Function Sets or queries the cursor type on the X-Y display.

Syntax
```
:CURSor:XY:TYPE {OFF|HORizontal|
VERTical|MARKer|HAVertical}
:CURSor:XY:TYPE?
```

Example
```
:CURSOR:XY:TYPE MARKER
:CURSOR:XY:TYPE? -> :CURSOR:XY:
TYPE MARKER
```

:CURSor:XY:VERTical?

Function Queries all V cursor settings for the X-Y display.

Syntax
```
:CURSor:XY:VERTical?
```

:CURSor:XY:VERTical:DX?

Function Queries all ΔX-axis settings. ΔX is between the V cursors on the X-Y display.

Syntax
```
:CURSor:XY:VERTical:DX?
```

:CURSor:XY:VERTical:DX:STATe

Function Sets or queries whether the ΔX-axis value between the V cursors on the X-Y display is displayed.

Syntax
```
:CURSor:XY:VERTical:DX:
STATe {<Boolean>}
:CURSor:XY:VERTical:DX:STATe?
```

Example
```
:CURSOR:XY:VERTICAL:DX:STATE ON
:CURSOR:XY:VERTICAL:DX:STATE? ->
:CURSOR:XY:VERTICAL:DX:STATE 1
```

:CURSor:XY:VERTical:DX:VALue?

Function Queries the ΔX-axis value between the V cursors on the X-Y display

Syntax
```
:CURSor:XY:VERTical:DX:VALue?
```

Example
```
:CURSOR:XY:VERTICAL:DX:VALUE? ->
:CURSOR:XY:VERTICAL:DX:
VALUE 300.00000E+00
```

:CURSor:XY:VERTical:POSition<x>

Function Sets or queries a V cursor position on the X-Y display.

Syntax
```
:CURSor:XY:VERTical:
POSition<x> {<NRf>}
:CURSor:XY:VERTical:POSition<x>?
<x> = 1, 2
```

Example
```
:CURSOR:XY:VERTICAL:POSITION1 -3
:CURSOR:XY:VERTICAL:POSITION1? ->
:CURSOR:XY:VERTICAL:POSITION1 -3.00
```

Description You can query cursor position information for cursors XY1 to XY8. Use the :CURSor:XY:VERTical:TRACe command to specify the waveform that you want to query.

:CURSor:XY:VERTical:TRACe

Function Sets or queries the source waveform that you want to measure using the V cursors on the X-Y display.

Syntax
```
:CURSor:XY:VERTical:TRACe {XY1|XY2|
XY3|XY4|XY5|XY6|XY7|XY8}
:CURSor:XY:VERTical:TRACe?
```

Example
```
:CURSOR:XY:VERTICAL:TRACE XY2
:CURSOR:XY:VERTICAL:TRACE? ->
:CURSOR:XY:VERTICAL:TRACE XY2
```

:CURSor:XY:VERTical:X<x>?

Function Queries all X-axis-value settings for a V cursor on the X-Y display.

Syntax
```
:CURSor:XY:VERTical:X<x>?
```

:CURSor:XY:VERTical:X<x>:STATe

Function Sets or queries whether the X-axis value for a V cursor on the X-Y display is displayed.

Syntax
```
:CURSor:XY:VERTical:X<x>:
STATe {<Boolean>}
:CURSor:XY:VERTical:X<x>:STATe?
<x> = 1, 2
```

Example
```
:CURSOR:XY:VERTICAL:X1:STATE ON
:CURSOR:XY:VERTICAL:X1:STATE? ->
:CURSOR:XY:VERTICAL:X1:STATE 1
```

:CURSor:XY:VERTical:X<x>:VALue?

Function Queries the X-axis value of a V cursor on the X-Y display.

Syntax
```
:CURSor:XY:VERTical:X<x>:VALue?
<x> = 1, 2
```

Example
```
:CURSOR:XY:VERTICAL:X1:VALUE? ->
:CURSOR:XY:VERTICAL:X1:
VALUE -150.00000E+00
```

:CURSor:WAIT?

Function While the DL850/DL850V is waiting for the specified :CURSor command to finish, it will wait the time specified by the :CURSor:WAIT? timeout value until it begins processing the subsequent command.

Syntax `:CURSor:WAIT? {<NRf>}`
`<NRf> = 1 to 36000 (timeout value, in units of 100 msec)`

Example `:CURSOR:WAIT? 100 -> :CURSOR:WAIT 1`

Description If the CURSor command that is being processed finishes within the specified timeout, this command will return 0. 1 is returned if the CURSor command does not finish within the specified timeout or if no CURSor command is being processed. Even if you make the timeout value long, 0 is returned as soon as the CURSor command finishes.

5.10 DISPlay Group

The commands in this group deal with the screen display. You can make the same settings and queries that you can by using keys such as the DISPLAY and UTILITY keys on the front panel.

:DISPlay?

Function Queries all display settings.

Syntax
```
:DISPlay?
```

:DISPlay:ACCumulate?

Function Queries all accumulated waveform display settings.

Syntax
```
:DISPlay:ACCumulate?
```

:DISPlay:ACCumulate:MODE

Function Sets or queries the accumulated waveform mode.

Syntax
```
:DISPlay:ACCumulate:MODE {OFF|ON|}
:DISPlay:ACCumulate:MODE?
```

Example
```
:DISPLAY:ACCUMULATE:MODE ON
:DISPLAY:ACCUMULATE:MODE? ->
:DISPLAY:ACCUMULATE:MODE 1
```

:DISPlay:ACCumulate:PERSistence

Function Sets or queries the number of times that waveforms are accumulated.

Syntax
```
:DISPlay:ACCumulate:
PERSistence {<NRf>|INFinity}
:DISPlay:ACCumulate:PERSistence?
<NRf> = 2 to 128 (in 2n steps)
```

Example
```
:DISPLAY:ACCUMULATE:PERSISTENCE 128
:DISPLAY:ACCUMULATE:PERSISTENCE? ->
:DISPLAY:ACCUMULATE:PERSISTENCE 128
```

:DISPlay:CINFormation (Channel Information)

Function Sets or queries the contents of the channel information display.

Syntax
```
:DISPlay:CINFormation {FULL|NARRow|
WIDE|WWSCale}
:DISPlay:CINFormation?
```

Example
```
:DISPLAY:CINFORMATION NARROW
:DISPLAY:CINFORMATION? -> :DISPLAY:
CINFORMATION NARROW
```

:DISPlay:CHANnel<x>:SCHannel<x>:COLor

Function Sets or queries a waveform color.

Syntax
- For channels that do not have sub channels or for math channels:
```
:DISPlay:{CHANnel<x>|MATH<x>}:
COLor {BLUE|BGReen|CYAN|DBLue|GRAY|
GREen|LBLue|LGReen|MAGenta|MGReen|
ORANge|PINK|PURPle|RED|SPINk|YELLow}
:DISPlay:{CHANnel<x>|MATH<x>}:
COLor?
```
- For channels that have sub channels:
```
:DISPlay:CHANnel<x>:SCHannel<x>:
COLor {BLUE|BGReen|CYAN|DBLue|GRAY|
GREen|LBLue|LGReen|MAGenta|MGReen|
ORANge|PINK|PURPle|RED|SPINk|YELLow}
:DISPlay:CHANnel<x>:SCHannel<x>:
COLor?
```

Example
```
:DISPLAY:CHANNEL1:COLOR YELLOW
:DISPLAY:CHANNEL1:COLOR? ->
:DISPLAY:CHANNEL1:COLOR YELLOW
```

Description The FFT1 and FFT2 waveforms are the same color as the MATH7 and MATH8 waveforms, respectively.

:DISPlay:COLor:BASecolor

Function Sets or queries the base color of the screen.

Syntax
```
:DISPlay:COLor:BASecolor {BLUE|GRAY}
:DISPlay:COLor:BASecolor?
```

Example
```
:DISPLAY:COLOR:BASECOLOR GRAY
:DISPLAY:COLOR:BASECOLOR? ->
:DISPLAY:COLOR:BASECOLOR GRAY
```

:DISPlay:DECimation

Function Sets or queries the number of dots (after decimation) that are used on the dot display.

Syntax
```
:DISPlay:DECimation {<NRf>}
:DISPlay:DECimation?
<NRf> = 2000 to 100000
```

Example
```
:DISPLAY:DECIMATION 2000
:DISPLAY:DECIMATION? -> :DISPLAY
:DECIMATION 2000
```

:DISPlay:ESIZe (Extra window Size)

Function Sets or queries the size of the extra window.

Syntax
```
:DISPlay:ESIZe {<NRf>|AUTO}
:DISPlay:ESIZe?
<NRf> = 0 to 8
```

Example
```
:DISPLAY:ESIZE AUTO
:DISPLAY:ESIZE? -> :DISPLAY:
ESIZE AUTO
```

:DISPlay:FORMat

Function Sets or queries the display format (the number of divisions in the vertical direction).

Syntax
```
:DISPlay:FORMat {<NRf>}
:DISPlay:FORMat?
<NRf> = 1, 2, 3, 4, 6, 8, 12, 16
```

Example
```
:DISPLAY:FORMAT 4
:DISPLAY:FORMAT? -> :DISPLAY:
FORMAT 4
```

:DISPlay:GRATicule

Function Sets or queries the grid type.

Syntax
```
:DISPlay:GRATicule {CROSshair|FRAMe|
GRID}
:DISPlay:GRATicule?
```

Example
```
:DISPLAY:GRATICULE GRID
:DISPLAY:GRATICULE? -> :DISPLAY:
GRATICULE GRID
```

:DISPlay:GROup<x1>?

Function Queries all display group settings.

Syntax
```
:DISPlay:GROup<x1>?
```

:DISPlay:GROup<x1>:TRACe<x2>?

Function Queries all source waveform settings for a display group.

Syntax
```
:DISPlay:GROup<x1>:TRACe<x2>?
```

:DISPlay:GROup<x1>:TRACe<x2>:SOURce

Function Sets or queries the source that is assigned to the specified source waveform of the specified display group.

Syntax
```
:DISPlay:GROup<x1>:TRACe<x2>:
SOURce {OFF|MATH<x>|<NRf>[,<NRf>]}
:DISPlay:GROup<x1>:TRACe<x2>:SOURce?
<x1> = 1 to 4, <x2> = 1 to 64
<NRf> = 1 to 16 (sub channel number)
MATH<x>'s <x> = 1 to 8
```

Example
```
:DISPLAY:GROUP1:TRACE1:SOURCE 1
:DISPLAY:GROUP1:TRACE1:SOURCE? ->
:DISPLAY:GROUP1:TRACE1:SOURCE 1
```

:DISPlay:GROup<x1>:TRACe<x2>:ZNUMber (Zone Number)

Function Sets or queries the zone number that displays the specified source waveform of the specified display group.

Syntax
```
:DISPlay:GROup<x1>:TRACe<x2>:
ZNUMber {<NRf>}
:DISPlay:GROup<x1>:TRACe<x2>:
ZNUMber?
<x1> = 1 to 4, <x2> = 1 to 64
<NRf> = 1 to 16
```

Example
```
:DISPLAY:GROUP1:TRACE1:ZNUMBER 1
:DISPLAY:GROUP1:TRACE1:ZNUMBER? ->
:DISPLAY:GROUP1:TRACE1:ZNUMBER 1
```

Description Regardless of the number of zones that have been set, you can always specify a number from 1 to 16 with this command.

:DISPlay:INTENsity?

Function Queries all intensity settings.

Syntax
```
:DISPlay:INTENsity?
```

:DISPlay:INTENsity:{CURSor|GRID|MARKer}

Function Sets or queries the intensity of a display item.

Syntax
```
:DISPlay:INTENsity:{CURSor|GRID|
MARKer} {<NRf>}
:DISPlay:INTENsity:{CURSor|GRID|
MARKer}?
<NRf> = 1 to 8
```

Example
```
:DISPLAY:INTENSITY:CURSOR 8
:DISPLAY:INTENSITY:CURSOR? ->
:DISPLAY:INTENSITY:CURSOR 8
```

:DISPlay:INTerpolate

Function Sets or queries the waveform interpolation method.

Syntax
```
:DISPlay:INTerpolate {LINE|OFF|
PULSe|SINE}
:DISPlay:INTerpolate?
```

Example
```
:DISPLAY:INTERPOLATE LINE
:DISPLAY:INTERPOLATE? -> :DISPLAY:
INTERPOLATE LINE
```

:DISPlay:SDGRoup (Select Display Group)

Function Sets or queries the number of the display group that is displayed.

Syntax
```
:DISPlay:SDGRoup {<NRf>}
:DISPlay:SDGRoup?
<NRf> = 1 to 4
```

Example
```
:DISPLAY:SDGROUP 1
:DISPLAY:SDGROUP? -> :DISPLAY:
SDGROUP 1
```

:DISPlay:MAPPing

Function Sets or queries the waveform mapping mode for when the display format is set to an option other than 1.

Syntax
```
:DISPlay:MAPPing {AUTO|USERdefine}
:DISPlay:MAPPing?
```

Example
```
:DISPLAY:MAPPING USERDEFINE
:DISPLAY:MAPPING? -> :DISPLAY:
MAPPING USERDEFINE
```

:DISPlay:SVALue (Scale VALUE)

Function Sets or queries whether scale values are displayed.

Syntax
```
:DISPlay:SVALue {<Boolean>}
:DISPlay:SVALue?
```

Example
```
:DISPLAY:SVALUE ON
:DISPLAY:SVALUE? -> :DISPLAY:
SVALUE 1
```

:DISPlay:TLABel (Trace LABEL)

Function Sets or queries whether waveform labels are displayed.

Syntax
```
:DISPlay:TLABel {<Boolean>}
:DISPlay:TLABel?
```

Example
```
:DISPLAY:TLABEL ON
:DISPLAY:TLABEL? -> :DISPLAY:
TLABEL 1
```

:DISPlay:LINDicator (Level Indicator)

Function Sets or queries whether the right indicator on the TY waveform display is displayed.

Syntax
```
:DISPlay:LINDicator {<Boolean>}
:DISPlay:LINDicator?
```

Example
```
:DISPLAY:LINDICATOR ON
:DISPLAY:LINDICATOR? -> :DISPLAY:
LINDICATOR 1
```

:DISPlay:MODE:EKEY (Esc Key)

Function Switches the screen display mode.
This command causes the DL850/DL850V to perform the same action as if its front panel ESC key had been pressed.

Syntax
```
:DISPlay:MODE:EKEY
```

Example
```
:DISPLAY:MODE:EKEY
```

5.11 EVENt Group

You can use the commands in this group to set data for and query data from events such as dual-capture and external-terminal-input events.

:EVENt?

Function Queries all event waveform settings.

Syntax `:EVENt?`

:EVENt:DCAPture[:MODE]

Function Sets or queries whether the event waveform's dual capture events are displayed.

Syntax `:EVENt:DCAPture[:MODE] {<Boolean>}`
`:EVENt:DCAPture[:MODE]?`

Example `:EVENT:DCAPTURE:MODE ON`
`:EVENT:DCAPTURE:MODE? -> :EVENT:DCAPTURE:MODE 1`

:EVENt:MEVent:MODE (Manual Event)

Function Sets or queries whether the event waveform's manual events are displayed.

Syntax `:EVENt:MEVent:MODE {<Boolean>}`
`:EVENt:MEVent:MODE?`

Example `:EVENT:MEVENT:MODE ON`
`:EVENT:MEVENT:MODE? -> :EVENT:MEVENT:MODE ON`

:EVENt:MEVent:EXECute

Function Manually inserts an event.

Syntax `:EVENt:MEVent:EXECute`

:EVENt:MEVent:SLOPe

Function Sets or queries the manual event (external event terminal) polarity.

Syntax `:EVENt:MEVent:SLOPe {FALL|RISE}`
`:EVENt:MEVent:SLOPe?`

Example `:EVENT:MEVENT:SLOPE RISE`
`:EVENT:MEVENT:SLOPE? -> :EVENT:MEVENT:SLOPE RISE`

5.12 FFT Group

The commands in this group deal with FFT analysis. You can make the same settings and queries that you can by using the SHIFT+MATH keys on the front panel.

:FFT?
Function Queries all FFT settings.
Syntax :FFT?

:FFT:AVERage?
Function Queries all FFT averaging settings.
Syntax :FFT:AVERage?

:FFT:AVERage:MODE
Function Sets or queries the FFT averaging mode.
Syntax :FFT:AVERage:MODE {EXPonent|LINear|OFF|PEAK}
:FFT:AVERage:MODE?
Example :FFT:AVERAGE:MODE PEAK
:FFT:AVERAGE:MODE? -> :FFT:AVERAGE:MODE PEAK

:FFT:AVERage:TYPE
Function Sets or queries the averaging domain.
Syntax :FFT:AVERage:TYPE {FREQuency|TIME}
:FFT:AVERage:TYPE?
Example :FFT:AVERAGE:TYPE FREQUENCY
:FFT:AVERAGE:TYPE? -> :FFT:AVERAGE:TYPE FREQUENCY

:FFT:AVERage:COUNt
Function Sets or queries the average count of linear averaging.
Syntax :FFT:AVERage:COUNt {<NRf>}
:FFT:AVERage:COUNt?
<NRf> = 2 to 128 (in 2n steps)
Example :FFT:AVERAGE:COUNT 32
:FFT:AVERAGE:COUNT? -> :FFT:AVERAGE:COUNT 32

:FFT:AVERage:EWEight
Function Sets or queries the attenuation constant of exponential averaging.
Syntax :FFT:AVERage:EWEight {<NRf>}
:FFT:AVERage:EWEight?
<NRf> = 2 to 256 (in 2n steps)
Example :FFT:AVERAGE:EWEIGHT 16
:FFT:AVERAGE:EWEIGHT? -> :FFT:AVERAGE:EWEIGHT 16

:FFT:POINt
Function Sets or queries the number of analysis source points of the FFT windows.
Syntax :FFT:POINt {<NRf>}
:FFT:POINt?
<NRf> = 1000, 2000, 5000, 10000, 20000, 50000, 100000
Example :FFT:POINT 1000
:FFT:POINT? -> :FFT:POINT 1000
Description The setting is the same for Window 1 and Window 2.

:FFT:STARt
Function Sets or queries the analysis-source start point of the FFT windows.
Syntax :FFT:STARt {<NRf>}
:FFT:STARt?
<NRf> = −5 to 5
Example :FFT:START -5
:FFT:START? -> :FFT:START -5.000000000000
Description The setting is the same for Window 1 and Window 2.

:FFT:WINDow?
Function Queries all FFT analysis settings.
Syntax :FFT:WINDow?

:FFT:WINDow:TYPE
Function Sets or queries the computation window type for FFT analysis.
Syntax :FFT:WINDow:TYPE {HANNing|RECTangle|FLATtop|EXPonential|HAMMing}
:FFT:WINDow:TYPE?
Example :FFT:WINDOW:TYPE HANNING
:FFT:WINDOW:TYPE? -> :FFT:WINDOW:TYPE HANNING
Description The setting is the same for Window 1 and Window 2.

:FFT:WINDow:EXPonential?
Function Queries all FFT exponential window settings.
Syntax :FFT:WINDow:EXPonential?

:FFT:WINDow:EXPonential:DRATe
Function Sets or queries the FFT exponential window's damping rate.
Syntax :FFT:WINDow:EXPonential:DRATe {<NRf>}
:FFT:WINDow:EXPonential:DRATe?
Example :FFT:WINDOW:EXPONENTIAL:DRATE 100
:FFT:WINDOW:EXPONENTIAL:DRATE? -> :FFT:WINDOW:EXPONENTIAL:DRATE 100
Description <NRf> = 1 to 100; the initial value and the setting are the same for Window 1 and Window 2.

:FFT:WINDow:EXPonential:FORCe<x>

Function Sets or queries the FFT exponential window's Force1 or Force2 value.

Syntax
```
:FFT:WINDow:EXPonential:
FORCe<x> {<NRf>}
:FFT:WINDow:EXPonential:FORCe<x>?
```

Example
```
:FFT:WINDOW:EXPONENTIAL:FORCE1 100
:FFT:WINDOW:EXPONENTIAL:FORCE1? ->
:FFT:WINDOW:EXPONENTIAL:FORCE1 100
```

Description `<NRf> = 1 to 100; the initial value and the setting are the same for Window 1 and Window 2.`

:FFT:WAVeform<x>?

Function Queries all FFT window settings.

`<x> = 1, 2`

Description `<x> = 2 is valid on models with the /G2 option.`

:FFT:WAVeform<x>:TYPE

Function Sets or queries an FFT window's analysis type.

Syntax
```
:FFT:WAVeform<x>:TYPE {LS|PS|PSD|
CS|TF|CH}
:FFT:WAVeform<x>:TYPE?
```
`<x> = 1, 2`

Example
```
:FFT:WAVEFORM1:TYPE PS
:FFT:WAVEFORM1:TYPE? -> :FFT:
WAVEFORM1:TYPE PS
```

Description This command is valid on models with the /G2 option.

:FFT:WAVeform<x>:STYPe (Sub Type)

Function Sets or queries an FFT window's analysis sub type.

Syntax
```
:FFT:WAVeform<x>:STYPe {REAL|IMAG|
MAG|PHASe|LOGMag}
:FFT:WAVeform<x>:STYPe?
```
`<x> = 1, 2`

Example
```
:FFT:WAVEFORM1:STYPE LOGMAG
:FFT:WAVEFORM1:STYPE? -> :FFT:
WAVEFORM1:STYPE LOGMAG
```

Description This command is valid on models with the /G2 option.

:FFT:WAVeform<x>:MODE

Function Sets or queries whether an FFT window is displayed.

Syntax
```
:FFT:WAVeform<x>:MODE {<Boolean>}
:FFT:WAVeform<x>:MODE?
```
`<x> = 1, 2`

Example
```
:FFT:WAVEFORM1:MODE ON
:FFT:WAVEFORM1:MODE? -> :FFT:
WAVEFORM1:MODE 0
```

Description `<x> = 2 is valid on models with the /G2 option.`

:FFT:WAVeform<x>:HAXis

Function Sets or queries an FFT window's horizontal-axis display method.

Syntax
```
:FFT:WAVeform<x>:HAXis {LINear|LOG}
:FFT:WAVeform<x>:HAXis?
```

Example
```
:FFT:WAVEFORM1:HAXIS LOG
:FFT:WAVEFORM1:HAXIS? -> :FFT:
WAVEFORM1:HAXIS LOG
```

:FFT:WAVeform<x>:VSCale?

Function Queries all FFT vertical scale settings.

Syntax `:FFT:WAVeform<x>:VSCale?`

:FFT:WAVeform<x>:VSCale:MODE (Vertical Scale Mode)

Function Sets or queries an FFT window's vertical scale mode.

Syntax
```
:FFT:WAVeform<x>:VSCale:MODE {AUTO|
MANual}
:FFT:WAVeform<x>:VSCale:MODE?
```

Example
```
:FFT:WAVEFORM1:VSCALE:MODE AUTO
:FFT:WAVEFORM1:VSCALE:MODE? -> :FFT:
WAVEFORM1:VSCALE:MODE AUTO
```

:FFT:WAVeform<x>:VSCale:CENTer

Function Sets or queries the center value of an FFT window's vertical scale.

Syntax
```
:FFT:WAVeform<x>:VSCale:
CENTer {<NRf>}
:FFT:WAVeform<x>:VSCale:CENTer?
<x> = 1, 2
<NRf> = -1E+30 to 1E+30
```

Example
```
:FFT:WAVEFORM1:VSCALE:CENTER -40
:FFT:WAVEFORM1:VSCALE:CENTER? ->
:FFT:WAVEFORM1:VSCALE:
CENTER -40.0000E+00
```

Description `<x> = 2 is valid on models with the /G2 option.`

:FFT:WAVeform<x>:VSCale:SENSitivity

Function Sets or queries the sensitivity of an FFT window's vertical scale.

Syntax
```
:FFT:WAVeform<x>:VSCALe:
SENSitivity {<NRf>}
:FFT:WAVeform<x>:VSCALe:SENSitivity?
<x> = 1, 2
<NRf> = 0 to 1E+30
```

Example
```
:FFT:WAVEFORM1:VSCALE:
SENSITIVITY 20.0000E+00
:FFT:WAVEFORM1:VSCALE:
SENSITIVITY? -> :FFT:WAVEFORM1:
VSCALE:SENSITIVITY 20.0000E+00
```

Description `<x> = 2 is valid on models with the /G2 option.`

:FFT:WAVeform<x>:SOURce<x>

Function Sets or queries an FFT window's FFT source channel.

Syntax
```
:FFT:WAVeform<x>:SOURce<x> {MATH<x>|<NRf> [,<NRf>]}
:FFT:WAVeform<x>:SOURce<x>?
WAVeform<x>'s <x> = 1, 2
SOURce<x>'s <x> = 1, 2
MATH<x>'s <x> = 1 to 6
First <NRf> = 1 to 16
Sub channel <NRf> = 1 to 16
```

Example
```
:FFT:WAVEFORM1:SOURCE1 1
:FFT:WAVEFORM1:SOURCE1? -> :FFT:WAVEFORM1:SOURCE1 1
```

Description This command returns "Math1" if no modules are installed.

5.13 FILE Group

The commands in this group deal with USB storage media, the internal hard disk, and external hard disks. You can perform the same operations and make the same settings and queries that you can by using the FILE key on the front panel.

:FILE?
Function Queries all storage media settings.
Syntax :FILE?

:FILE:COPY:ABORt
Function Aborts file copying.
Syntax :FILE:COPY:ABORt
Example :FILE:COPY:ABORT

:FILE:COPY:CDIRectory
Function Changes the file copy destination directory.
Syntax :FILE:COPY:CDIRectory {<String>}
<String> = Directory name
Example :FILE:COPY:CDIRECTORY "NO_1"
Description <String> specifies a relative path.

:FILE:COPY[:EXECute]
Function Executes file copying. This is an overlap command.
Syntax :FILE:COPY[:EXECute] {<String>}
<String> = The file name, including its extension.
Example :FILE:COPY:EXECUTE" DATA.PNG"

:FILE:COPY:PATH?
Function Queries the file copy destination directory.
Example :FILE:COPY:PATH? -> :FILE:COPY:PATH" PATH=USB/UTIL"

:FILE:DELete
Function Deletes files. This is an overlap command.
Syntax :FILE:DELete {<String>}
Example :FILE:DELETE" CASE1.WDF"
Description
- Use the FILE[:DIRectory]:DRIVe command to select the target medium.
- Use the FILE[:DIRectory]:CDIRectory command to select the directory that contains the file that you want to delete.
- Include the extension when you specify the file name.
- If you specify a directory, the directory will be deleted.

:FILE[:DIRectory]:CDIRectory (Change Directory)
Function Changes the current directory on the storage medium.
Syntax :FILE[:DIRectory]:
CDIRectory {<String>}
<String> = Up to 16 characters
Example :FILE:DIRECTORY:CDIRECTORY" NO_1"

:FILE[:DIRectory]:DRIVe
Function Sets the storage medium to perform file operations on.
Syntax :FILE[:DIRectory]:DRIVe {IFMemory|HD,<NRf>|NETWork|SD|USB ,<NRf>}
Example :FILE:DIRECTORY:DRIVE HD
Description Specify HD for the internal hard disk.
<NRf> = This can be omitted if the disk is not partitioned. For the DL850/DL850V internal hard disk, omit this value or specify 0.
Specify SD for the SD memory card.
Specify NETWork for a network drive.
Specify USB for a USB storage medium.
<NRf>: 0 to LUN or the partition number (can be omitted)
On the DL850/DL850V, this number is decided by the order that USB storage media are inserted into the device.
Partition numbers:
Internal hard disk: 0 to 2
USB: 0 to 3

:FILE[:DIRectory]:FREE?
Function Queries the free space on the medium that is being operated on in bytes.
Syntax :FILE[:DIRectory]:FREE?
Example :FILE:DIRECTORY:FREE? -> :FILE:DIRECTORY:FREE 3.7567939E+09

:FILE[:DIRectory]:MDIRectory (Make Directory)
Function Creates a directory in the current directory.
Syntax :FILE[:DIRectory]:
MDIRectory {<String>}
Example :FILE:DIRECTORY:MDIRECTORY "NO_1"
Description <String> specifies a relative path.

:FILE[:DIRectory]:PATH?

Function Queries the current directory.

Syntax `:FILE[:DIRectory]:PATH?`

Example `:FILE:DIRECTORY:PATH? -> :FILR:PATH" Path=HD,1/20090506"`

:FILE:LOAD:{BINary|SETup|SNAP}:ABORt

Function Aborts the loading of various types of data.

Syntax `:FILE:LOAD:{BINary|SETup|SNAP}:ABORt`

Example `:FILE:LOAD:SNAP:ABOR`

:FILE:LOAD:{BINary|SETup|SNAP}[:EXECute]

Function Executes the loading of various types of data. This is an overlap command.

Syntax `:FILE:LOAD:{BINary|SETup|SNAP}[:EXECute] {<Filename>}`

Example `:FILE:LOAD:SETUP "CASE1"`

Description Do not include the extension when you specify <Filename>.

:FILE:SAVE?

Function Queries all file name settings for saving files.

Syntax `:FILE:SAVE?`

:FILE:SAVE:ANAMing

Function Sets or queries the auto naming mode for saving files.

Syntax `:FILE:SAVE:ANAMing {DATE|NUMBering|OFF}`
`:FILE:SAVE:ANAMing?`

Example `:FILE:SAVE:ANAMING DATE`
`:FILE:SAVE:ANAMING? -> :FILE:SAVE:ANAMING DATE`

Description DATE: Date and time
NUMBering: Numbering
OFF: The file name that you saved with the FILE:SAVE:NAME command

:FILE:SAVE:{ASCii|BINary|FLOat}?

Function Queries all the settings related to the saving of a specific type of file.

Syntax `:FILE:SAVE:{ASCii|BINary|FLOat}?`

:FILE:SAVE:{ASCii|BINary|FLOat|MEASure|SETup|SNAP}:ABORt

Function Aborts the saving of a specific type of file.

Syntax `:FILE:SAVE:{ASCii|BINary|FLOat|MEASure|SETup|SNAP}:ABORt`

Example `:FILE:SAVE:SETUP:ABORT`

:FILE:SAVE:{ASCii|BINary|FLOat|MEASure|SETup|SNAP}[:EXECute]

Function Executes the saving of a specific type of file. This is an overlap command.

Syntax `:FILE:SAVE:{ASCii|BINary|FLOat|MEASure|SETup|SNAP}[:EXECute]`

Example `:FILE:SAVE:SETUP:EXECUTE`

:FILE:SAVE:{ASCii|BINary|FLOat}:HISTory

Function Sets or queries what waveforms the history memory feature will save for a specific type of data.

Syntax `:FILE:SAVE:{ASCii|BINary|FLOat}:HISTORY {ONE|ALL}`
`:FILE:SAVE:{ASCii|BINary|FLOat}:HISTORY?`

Example `:FILE:SAVE:BINARY:HISTORY ALL`
`:FILE:SAVE:BINARY:HISTORY? -> :FILE:SAVE:BINARY:HISTORY ALL`

:FILE:SAVE:{ASCii|BINary|FLOat}:RANGe

Function Sets or queries the save range for a specific type of data.

Syntax `:FILE:SAVE:{ASCii|BINary|FLOat}:RANGe {MAIN|Z1|Z2|CURSor}`
`:FILE:SAVE:{ASCii|BINary|FLOat}:RANGe?`

Example `:FILE:SAVE:BINARY:RANGE CURSOR`
`:FILE:SAVE:BINARY:RANGE? -> :FILE:SAVE:BINARY:RANGE CURSOR`

:FILE:SAVE:{ASCii|BINary|FLOat}:CRANge<x>

Function Sets or queries the cursor position for when data is to be saved using a cursor range.

Syntax `:FILE:SAVE:{ASCii|BINary|FLOat}:CRANGe<x> {<NRf>}`
`:FILE:SAVE:{ASCii|BINary|FLOat}:CRANGe<x>?`
`<x> = 1, 2`
`<NRf> = Resolution (depends on the measurement length)`

:FILE:SAVE:{ASCii|BINary|FLOat}:STRace:{CHANnel<x>|MATH<x>}

Function Sets or queries the waveform that will be saved for a specific type of data.

Syntax `:FILE:SAVE:{ASCii|BINary|FLOat}:STRace:{CHANnel<x>|MATH<x>} {<Boolean>}`
`:FILE:SAVE:{ASCii|BINary|FLOat}:STRace:{CHANnel<x>|MATH<x>}?`

Example `:FILE:SAVE:BINARY:STRACE:CHANNEL1 ON`
`:FILE:SAVE:BINARY:STRACE:CHANNEL1? -> :FILE:SAVE:BINARY:STRACE:CHANNEL1 1`

Description When the Trace All setting is set to OFF, the source waveforms of waveforms that are displayed and that have this setting set to ON will be saved.
You cannot specify waveforms by sub channel.

:FILE:SAVE:{ASCii|BINary|FLOat}:TALL (Trace All)

Function Sets or queries the selection method for the waveforms that will be saved for a specific type of data.

Syntax
```
:FILE:SAVE:{ASCii|BINary|FLOat}:
TALL {<Boolean>}
:FILE:SAVE:{ASCii|BINary|FLOat}:
TALL?
```

Example
```
:FILE:SAVE:BINARY:TALL ON
:FILE:SAVE:BINARY:TALL? -> :FILE:
SAVE:BINARY:TALL 1
```

Description ON: All channels whose waveforms are displayed will be saved.
OFF: The channels whose waveforms are displayed and whose waveforms are set to be saved will be saved.

:FILE:SAVE:ASCii:INTerval

Function Sets or queries the data removal interval that is used when saving data in ASCII format.

Syntax
```
:FILE:SAVE:ASCii:INTerval {<NRf>|
OFF}
:FILE:SAVE:ASCii:INTerval?
```

Example
```
:FILE:SAVE:ASCII:INTERVAL 5
:FILE:SAVE:ASCII:INTERVAL? -> :FILE:
SAVE:ASCII:INTERVAL 5
<NRf> = 5, 10, 20, 50, 100, 200,
500, 1000, 2000, 5000
```

:FILE:SAVE:ASCii:TINFormation

Function Sets or queries whether time information is included when saving data in ASCII format.

Syntax
```
:FILE:SAVE:ASCii:
TINFormation {<Boolean>}
:FILE:SAVE:ASCii:TINFormation?
```

Example
```
:FILE:SAVE:ASCII:TINFORMATION ON
:FILE:SAVE:ASCII:TINFORMATION? ->
:FILE:SAVE:ASCII:TINFORMATION 1
```

:FILE:SAVE:ASCii:DPOint

Function Sets or queries the type of decimal point that is used when saving data in ASCII format.

Syntax
```
:FILE:SAVE:ASCii:DPOint {POINt|
COMMa}
:FILE:SAVE:ASCii:DPOint?
```

Example
```
:FILE:SAVE:ASCII:DPOINT POINT
:FILE:SAVE:ASCII:DPOINT? -> :FILE:
SAVE:ASCII:DPOINT POINT
```

:FILE:SAVE:ASCii:SCHannel

Function Sets or queries the writing method of sub channel data when saving data in ASCII format.

Syntax
```
:FILE:SAVE:ASCii:
SCHannel {SUPPlement|SPACe}
:FILE:SAVE:ASCii:SCHannel?
```

Example
```
:FILE:SAVE:ASCII:SCHANNEL SPACE
:FILE:SAVE:ASCII:SCHANNEL? -> :FILE:
SAVE:ASCII:SCHANNEL SPACE
```

:FILE:SAVE:ASCii:EXTension

Function Sets or queries the file name extension that is used when saving data in ASCII format.

Syntax
```
:FILE:SAVE:ASCii:EXTension {CSV|
MATLab}
:FILE:SAVE:ASCii:EXTension?
```

Example
```
:FILE:SAVE:ASCII:EXTENSION CSV
:FILE:SAVE:ASCII:EXTENSION? ->
:FILE:SAVE:ASCII:EXTENSION CSV
```

:FILE:SAVE:BINary:COMPression

Function Sets or queries whether waveform data is compressed during saving.

Syntax
```
:FILE:SAVE:BINary:
COMPression {<Boolean>}
:FILE:SAVE:BINary:COMPression?
```

Example
```
:FILE:SAVE:BINARY:COMPRESSION ON
:FILE:SAVE:BINARY:COMPRESSION? ->
:FILE:SAVE:BINARY:COMPRESSION 1
```

Description This command is valid when you have set the waveform to save to the full range of waveforms on the main screen. Data can only be saved in binary format.

:FILE:SAVE:COMMent

Function Sets or queries the comment that will be saved.

Syntax
```
:FILE:SAVE:COMMent {<String>}
:FILE:SAVE:COMMent?
<String> = Up to 250 characters
```

Example
```
:FILE:SAVE:COMMENT "WAVEFORM_1"
:FILE:SAVE:COMMENT? -> :FILE:SAVE:
COMMENT "WAVEFORM_1"
```

Description You can only use the characters and symbols on the keyboard that appears on the DL850/DL850V screen. “Ω” is “1EH” and “μ” is “1FH” expressed in ASCII code.

:FILE:SAVE:NAME

Function Sets or queries the name of the file that will be saved.

Syntax
```
:FILE:SAVE:NAME <Filename>
:FILE:SAVE:NAME?
```

Example
```
:FILE:SAVE:NAME "CASE1"
:FILE:SAVE:NAME? -> :FILE:SAVE:
NAME "CASE1"
```

5.14 GONogo Group

The commands in this group deal with GO/NO-GO determination. You can make the same settings and queries (of settings and measured values) that you can by using the SHIFT+MEASURE keys on the front panel.

If the selectable range of the time axis is "<NRf> = –5 to 5div," the selectable range varies depending on settings such as the record length. For details, see "Notes about Cursor Measurement" in chapter 9 of IM DL850-01EN.

:GONogo?

Function Queries all GO/NO-GO determination settings.

Syntax
```
:GONogo?
```

:GONogo:ACONdition

Function Sets or queries the GO/NO-GO determination-action condition.

Syntax
```
:GONogo:ACONdition {ALWays|FAILure|SUCCess}
:GONogo:ACONdition?
```

Example
```
:GONOGO:ACONDITION ALWAYS
:GONOGO:ACONDITION? -> :GONOGO:ACONDITION ALWAYS
```

:GONogo:ACTion?

Function Queries all settings for the action that is performed when the condition is met and the settings for the condition itself.

Syntax
```
:GONogo:ACTion?
```

:GONogo:ACTion:BUZZer

Function Sets or queries whether a beep is sounded when the condition is met.

Syntax
```
:GONogo:ACTion:BUZZer {<Boolean>}
:GONogo:ACTion:BUZZer?
```

Example
```
:GONOGO:ACTION:BUZZER ON
:GONOGO:ACTION:BUZZER? -> :GONOGO:ACTION:BUZZER 1
```

:GONogo:ACTion:HCOPy (HardCOPY)

Function Sets or queries whether a screen capture is printed from the built-in printer when the condition is met.

Syntax
```
:GONogo:ACTion:HCOPy {<Boolean>}
:GONogo:ACTion:HCOPy?
```

Example
```
:GONOGO:ACTION:HCOPY ON
:GONOGO:ACTION:HCOPY? -> :GONOGO:ACTION:HCOPY 1
```

:GONogo:ACTion:IMAGe?

Function Queries all settings for the screen capture that is saved when the condition is met.

Syntax
```
:GONogo:ACTion:IMAGe?
```

:GONogo:ACTion:IMAGe[:MODE]

Function Sets or queries whether a screen capture is saved when the condition is met.

Syntax
```
:GONogo:ACTion:IMAGe[:MODE] {<Boolean>}
:GONogo:ACTion:IMAGe:MODE?
```

Example
```
:GONOGO:ACTION:IMAGE:MODE ON
:GONOGO:ACTION:IMAGE:MODE? -> :GONOGO:ACTION:IMAGE:MODE 1
```

:GONogo:ACTion:IMAGe:CDIRectory

Function Sets the current directory where the screen capture is saved to when the condition is met.

Syntax
```
:GONogo:ACTion:IMAGe:CDIRectory {<String>}
:GONogo:ACTion:IMAGe:CDIRectory?
<String> = Directory name
```

Example
```
:GONOGO:ACTION:IMAGE:CDIRECTORY "CASE1"
:GONOGO:ACTION:IMAGE:CDIRECTORY? -> :GONOGO:ACTION:IMAGE:CDIRECTORY "CASE1"
```

:GONogo:ACTion:IMAGe:DRIVe

Function Sets or queries the medium that the screen capture is saved to when the condition is met.

Syntax
```
:GONogo:ACTion:IMAGe:
DRIVe {HD,<NRf>|NETWork|SD|
USB,<NRf>}
:GONogo:ACTion:IMAGe:DRIVe?
```

Example
```
:GONOGO:ACTION:IMAGE:DRIVE HD
:GONOGO:ACTION:IMAGE:DRIVE? ->
:GONOGO:ACTION:IMAGE:DRIVE HD,0
```

Description
Specify HD for the internal hard disk.
<NRf> = This can be omitted if the disk is not partitioned. For the DL850/DL850V internal hard disk, omit this value or specify 0.
Specify SD for the SD memory card.
Specify NETWork for a network drive.
Specify USB for a USB storage medium.
First <NRf>: 0 to LUN or the partition number (can be omitted)
On the DL850/DL850V, this number is decided by the order that USB storage media are inserted into the device.
Partition numbers:
Internal hard disk: 0 to 2
USB: 0 to 3

:GONogo:ACTion:IMAGe:PATH?

Function Queries the path on the storage medium that a screen capture is saved to when the condition is met.

Syntax
```
:GONogo:ACTion:IMAGe:PATH?
```

Example
```
:GONOGO:ACTION:IMAGE:PATH? ->
:GONOGO:ACTION:IMAGE:
PATH "PATH=USB/UTIL"
```

:GONogo:ACTion:MAIL?

Function Queries all settings for sending e-mail when the condition is met.

Syntax
```
:GONogo:ACTion:MAIL?
```

:GONogo:ACTion:MAIL:COUNt

Function Sets or queries the number of times that e-mail is sent when the condition is met.

Syntax
```
:GONogo:ACTion:MAIL:COUNt {<NRf>}
:GONogo:ACTion:MAIL:COUNt?
<NRf> = 1 to 1000
```

Example
```
:GONOGO:ACTION:MAIL:COUNT 100
:GONOGO:ACTION:MAIL:COUNT? ->
:GONOGO:ACTION:MAIL:COUNT 100
```

:GONogo:ACTion:MAIL:MODE

Function Sets or queries whether e-mail is sent when the condition is met.

Syntax
```
:GONogo:ACTion:MAIL:MODE {<Boolean>}
:GONogo:ACTion:MAIL:MODE?
```

Example
```
:GONOGO:ACTION:MAIL:MODE ON
:GONOGO:ACTION:MAIL:MODE? ->
:GONOGO:ACTION:MAIL:MODE 1
```

:GONogo:ACTion:SAVE[:MODE]

Function Sets or queries whether waveforms are saved to the storage medium when the condition is met.

Syntax
```
:GONogo:ACTion:SAVE:MODE {<Boolean>}
:GONogo:ACTion:SAVE:MODE?
```

Example
```
:GONOGO:ACTION:SAVE:MODE ON
:GONOGO:ACTION:SAVE:MODE? ->
:GONOGO:ACTION:SAVE:MODE 1
```

:GONogo:ACTion:SAVE:CDIRectory

Function Sets the current directory on the storage medium where waveform data is saved to when the condition is met.

Syntax
```
:GONogo:ACTion:SAVE:
CDIRectory {<String>}
:GONogo:ACTion:SAVE:CDIRectory?
```

Example
```
:GONOGO:ACTION:SAVE:
CDIRECTORY "CASE1"
:GONOGO:ACTION:SAVE:CDIRECTORY? ->
:GONOGO:ACTION:SAVE:
CDIRECTORY "CASE1"
```

:GONogo:ACTion:SAVE:DRIVe

Function Sets the storage medium that waveform data is saved to when the condition is met.

Syntax
```
:GONogo:ACTion:SAVE:DRIVe {HD,<NRf>|NETWork|SD|USB,<NRf>}
:GONogo:ACTion:SAVE:DRIVe?
```

Example
```
:GONOGO:ACTION:SAVE:DRIVE HD
:GONOGO:ACTION:SAVE:DRIVE? ->
:GONOGO:ACTION:SAVE:DRIVE HD,0
```

Description
Specify HD for the internal hard disk.
<NRf> = This can be omitted if the disk is not partitioned. For the DL850/DL850V internal hard disk, omit this value or specify 0.
Specify SD for the SD memory card.
Specify NETWork for a network drive.
Specify USB for a USB storage medium.
First <NRf>: 0 to LUN or the partition number (can be omitted)
On the DL850/DL850V, this number is decided by the order that USB storage media are inserted into the device.
Partition numbers:
Internal hard disk: 0 to 2
USB: 0 to 3

:GONogo:ACTion:SAVE:PATH?

Function Queries the path on the storage medium that waveform data is saved to when the condition is met.

Syntax
```
:GONogo:ACTion:SAVE:PATH?
```

Example
```
:GONOGO:ACTION:SAVE:PATH? ->
:GONOGO:ACTION:SAVE:
PATH "PATH=USB/UTIL"
```

:GONogo:ACTion:SAVE:TYPE

Function Sets or queries the file format that waveforms are saved as on the storage medium when the condition is met.

Syntax
```
:GONogo:ACTion:SAVE:TYPE {ASCii|BINary|FLOat}
:GONogo:ACTion:SAVE:TYPE?
```

Example
```
:GONOGO:ACTION:SAVE:TYPE BINARY
:GONOGO:ACTION:SAVE:TYPE? ->
:GONOGO:ACTION:SAVE:TYPE BINARY
```

:GONogo:COUNt?

Function Queries the number of GO/NO-GO determinations that were performed.

Syntax
```
:GONogo:COUNt?
```

Example
```
:GONOGO:COUNT? -> :GONOGO:COUNT 10
```

:GONogo:LOGic

Function Sets or queries the GO/NO-GO determination logic.

Syntax
```
:GONogo:LOGic {AND|OR}
:GONogo:LOGic?
```

Example
```
:GONOGO:LOGIC AND
:GONOGO:LOGIC? -> :GONOGO:LOGIC AND
```

:GONogo:MODE

Function Sets or queries the GO/NO-GO determination mode.

Syntax
```
:GONogo:MODE {OFF|PARameter|ZONE}
:GONogo:MODE?
```

Example
```
:GONOGO:MODE ZONE
:GONOGO:MODE? -> :GONOGO:MODE ZONE
```

:GONogo:NGCount?

Function Queries the GO/NO-GO determination NO-GO count.

Syntax
```
:GONogo:NGCount?
```

Example
```
:GONOGO:NGCOUNT? -> :GONOGO:
NGCOUNT 10
```

:GONogo:PARameter?

Function Queries all parameter determination settings.

Syntax
```
:GONogo:PARameter?
```

:GONogo:PARameter:ITEM<x>?

Function Queries all settings for the specified waveform parameter for parameter determination.

Syntax
```
:GONogo:PARameter:ITEM<x>?
<x> = 1 to 16
```

:GONogo:PARameter:ITEM<x>:CAUSe?

Function Queries whether the specified waveform parameter for parameter determination is the cause of a NO-GO judgment.

Syntax
```
:GONogo:PARameter:ITEM<x>:CAUSe?
<x> = 1 to 16
```

Example
```
:GONOGO:PARAMETER:ITEM1:CAUSE? ->
:GONOGO:PARAMETER:ITEM1:CAUSE 1
```

Description When the parameter is the cause of a NO-GO result, the DL850/DL850V returns 1. Otherwise, the DL850/DL850V returns 0.

:GONogo:PARameter:ITEM<x>:MODE

Function Sets or queries the specified waveform parameter's reference condition for parameter determination.

Syntax
```
:GONogo:PARameter:ITEM<x>:MODE {OFF|IN|OUT}
:GONogo:PARameter:ITEM<x>:MODE?
<x> = 1 to 16
```

Example
```
:GONOGO:PARAMETER:ITEM1:MODE IN
:GONOGO:PARAMETER:ITEM1:MODE? ->
:GONOGO:PARAMETER:ITEM1:MODE IN
```

:GONogo:PARameter:ITEM<x>:TRACe

Function Sets or queries the specified waveform parameter's source waveform for parameter determination.

Syntax
```
:GONogo:PARameter:ITEM<x>:
TRACe {<NRf>}
:GONogo:PARameter:ITEM<x>:TRACe?
<x> = 1 to 16
<NRf> = 1 to 16  Channel number
```

Example
```
:GONOGO:PARAMETER:ITEM1:TRACE 1
:GONOGO:PARAMETER:ITEM1:TRACE? ->
:GONOGO:PARAMETER:ITEM1:TRACE 1
```

:GONogo:PARameter:ITEM<x>:TYPE?

Function Queries, for parameter determination, the specified waveform parameter's measurement item and upper and lower limits.

Syntax
```
:GONogo:PARameter:ITEM<x>:TYPE?
<x> = 1 to 16
```

Example
```
:GONOGO:PARAMETER:ITEM1:TYPE? ->
:GONOGO:PARAMETER:ITEM1:TYPE:
MAXIMUM 1.10000E+00,1.00000E+00
```

:GONogo:PARameter:ITEM<x>:TYPE:<Parameter>

Function Sets or queries the upper and lower limits of the measurement item for the specified waveform parameter.

Syntax
```
:GONogo:PARameter:ITEM<x>:
TYPE:<Parameter> {<{Voltage|
DONTcare}>,<{Voltage|DONTcare}>|
<{Current|DONTcare},<{Current|
DONTcare}>|<{Time|DONTcare},<{Time|
DONTcare}>|<{Frequency|DONTcare},
<{Frequency|DONTcare}>| <{<NRf>|
DONTcare}>,<{<NRf>|DONTcare}>}
<x> = 1 to 16
<Parameter> = {AMPLitude|AVERage|
AVGFreq|AVGPeriod|BWIDth1|BWIDth2|
DELay|DUTYcycle|FALL|FREQuency|HIGH|LOW
|MAXimum|MIDDle|MINimum|NOVershoot|NWID
th|PERiod|PNUMber|POVershoot|
PTOPeak|PWIDth|RISE|RMS|SDEViation|
TY1Integ|TY2Integ}
```

:GONogo:PARameter:ITEM<x>:VALue?

Function Queries the measured value of the specified waveform parameter.

Syntax
```
:GONogo:PARameter:ITEM<x>:VALue?
<x> = 1 to 16
```

:GONogo:REMote

Function Sets or queries the remote mode of GO/NO-GO determination.

Syntax
```
:GONogo:REMote {<Boolean>}
:GONogo:REMote?
```

Example
```
:GONOGO:REMOTE ON
:GONOGO:REMOTE? -> :GONOGO:REMOTE 1
```

:GONogo:SEQuence

Function Sets or queries the action mode of GO/NO-GO determination.

Syntax
```
:GONogo:SEQuence {CONTinue|SINGle}
:GONogo:SEQuence?
```

Example
```
:GONOGO:SEQUENCE CONTINUE
:GONOGO:SEQUENCE? -> :GONOGO:
SEQUENCE CONTINUE
```

:GONogo:TRANge

Function Sets or queries the determination range of GO/NO-GO determination.

Syntax
```
:GONogo:TRANge {<NRf>,<NRf>}
:GONogo:TRANge?
<NRf> = -5 to 5 div
```

Example
```
:GONOGO:TRANGE -5, 5
:GONOGO:TRANGE? -> :GONOGO:
TRANGE -5.000000000000,
5.000000000000
```

:GONogo:ZONE?

Function Queries all waveform zone determination settings.

Syntax
```
:GONogo:ZONE?
```

:GONogo:ZONE:PATTern<x>?

Function Queries all settings for the specified determination pattern for waveform zone determination.

Syntax
```
:GONogo:ZONE:PATTern<x>?
```

:GONogo:ZONE:PATTern<x>:CAUSe?

Function Queries whether the specified determination pattern for waveform zone determination is the cause of a NO-GO judgment.

Syntax
```
:GONogo:ZONE:PATTern<x>:CAUSe?
<x> = 1 to 16
```

Example
```
:GONOGO:ZONE:PATTERN1 CAUSE? ->
:GONOGO:ZONE:PATTERN1 CAUSE 0
```

Description When the determination pattern is the cause of a NO-GO result, the DL850/DL850V returns 1. Otherwise, the DL850/DL850V returns 0.

:GONogo:ZONE:PATTern<x>:MODE

Function Sets or queries the specified determination pattern's reference condition for waveform zone determination.

Syntax
```
:GONogo:ZONE:PATTern<x>:MODE {IN|
OUT|OFF}
:GONogo:ZONE:PATTern<x>:MODE?
<x> = 1 to 16
```

Example
```
:GONOGO:ZONE:PATTERN1:MODE IN
:GONOGO:ZONE:PATTERN1:MODE? ->
:GONOGO:ZONE:PATTERN1:MODE IN
```

:GONogo:ZONE:PATTern<x>:TRACe

Function Sets or queries the specified determination pattern's determination waveform for waveform zone determination.

Syntax
```
:GONogo:ZONE:PATTern<x>:
TRACe {<NRf>}
:GONogo:ZONE:PATTern<x>:TRACe?
<x> = 1 to 16
```

Example
```
:GONOGO:ZONE:PATTERN1:TRACE 1
:GONOGO:ZONE:PATTERN1:TRACE? ->
:GONOGO:ZONE:PATTERN1:TRACE 1
```

:GONogo:ZONE:PATTern<x>:ZONE

Function Sets or queries the specified determination pattern's source waveform zone data for waveform zone determination.

Syntax
```
:GONogo:ZONE:PATTern<x>:ZONE {<NRf>}
:GONogo:ZONE:PATTern<x>:ZONE?
<x> = 1 to 16
<NRf> = 0 to 5
```

Example
```
:GONOGO:ZONE:PATTERN1:ZONE 1
:GONOGO:ZONE:PATTERN1:ZONE? ->
:GONOGO:ZONE:PATTERN1:ZONE 1
```

:GONogo:WAIT?

Function Waits for the completion of GO/NO-GO determination by using a timeout value.

Syntax
```
:GONogo:WAIT? {<NRf>}
<NRf> = 1 to 864000 (in units of 100
ms)
```

Example
```
Setting the timeout value to 5
seconds
:GONOGO:WAIT? 50 -> 0
```

Description The DL850/DL850V returns 0 if the operation finishes before the timer expires and returns 1 if the timer expires.

5.15 HCOPy Group

The commands in this group deal with printing screen captures from the internal printer and other devices. You can perform the same operations and make the same settings and queries that you can by using the PRINT MENU key on the front panel.

:HCOPy?

Function Queries all screen capture data output settings.

Syntax
```
:HCOPy?
```

:HCOPy:COMMent

Function Sets or queries the screen comment.

Syntax
```
:HCOPy:COMMent {<String>}
:HCOPy:COMMent?
<String> = Up to 26 characters
```

Example
```
:HCOPY:COMMENT "ABC"
:HCOPY:COMMENT? -> :HCOPY:
COMMENT "ABC"
```

:HCOPy:DIRection

Function Sets or queries the data output destination.

Syntax
```
:HCOPy:DIRection {PRINter|
NETPrinter}
:HCOPy:DIRection?
```

Example
```
:HCOPY:DIRECTION FILE
:HCOPY:DIRECTION? -> :HCOPY:
DIRECTION FILE
```

:HCOPy:EXECute

Function Executes data output.

Syntax
```
:HCOPy:EXECute
```

:HCOPy:NETPrint?

Function Queries all network printer output settings.

Syntax
```
:HCOPy:NETPrint?
```

:HCOPy:NETPrint:TONE

Function Sets or queries whether data will be printed in color from the network printer.

Syntax
```
:HCOPy:NETPrint:TONE {<Boolean>}
:HCOPy:NETPrint:TONE?
```

Example
```
:HCOPY:NETPRINT:TONE ON
:HCOPY:NETPRINT:TONE? -> :HCOPY:
NETPRINT:TONE 1
```

:HCOPy:NETPrint:TYPE

Function Sets or queries the network printer output command type.

Syntax
```
:HCOPy:NETPrint:TYPE {EINKjet|
HINKjet|HLASer}
:HCOPy:NETPrint:TYPE?
```

Example
```
:HCOPY:NETPRINT:TYPE EINKJET
:HCOPY:NETPRINT:TYPE? -> :HCOPY:
NETPRINT:TYPE EINKJET
```

5.16 HISTory Group

The commands in this group deal with loading data from history waveforms. You can perform the same operations and make the same settings and queries that you can by using the HISTORY key on the front panel.

:HISTory?
Function Queries all of the settings for the history feature.
Syntax :HISTory?

:HISTory:ABORt
Function Aborts the history search.
Syntax :HISTory:ABORt
Example :HISTORY:ABORT

:HISTory:CLEar
Function Clears the data of all history waveforms.
Syntax :HISTory:CLEar
Example :HISTORY:CLEAR

:HISTory:DISPlay
Function Sets or queries the history start and end numbers that will be displayed.
Syntax :HISTory:DISPlay {<NRf>,<NRf>}
:HISTory:DISPlay?
<NRf> = 0 to −4999
Example :HISTORY:DISPLAY -1,-2
:HISTORY:DISPLAY? -> :HISTORY:DISPLAY -1,-2
Description The settable values vary depending on the memory model and the acquisition conditions.

:HISTory:DMODe (Display MODE)
Function Sets or queries the history waveform display mode.
Syntax :HISTory:DMODe {ONE|ALL|AVE}
:HISTory:DMODe?
Example :HISTORY:DMODE ONE
:HISTORY:DMODE? -> :HISTORY:DMODE ONE

:HISTory:EXECute
Function Executes the history waveform search. This is an overlap command.
Syntax :HISTory:EXECute

:HISTory:PARameter?
Function Queries all history-waveform parameter-search settings.
Syntax :HISTory:PARameter?

:HISTory:PARameter:ITEM<x>?
Function Queries all the specified parameter's settings for history-waveform parameter searches.
Syntax :HISTory:PARameter:ITEM<x>?
<x> = 1 to 4

:HISTory:PARameter:ITEM<x>:CONDition
Function Sets or queries the specified parameter's reference condition for history-waveform parameter searches.
Syntax :HISTory:PARameter:ITEM<x>:
CONDition {OFF|IN|OUT}
<x> = 1 to 4
Example :HISTORY:PARAMETER:ITEM1:
CONDITION IN
:HISTORY:PARAMETER:ITEM1:CONDITION?
-> :HISTORY:PARAMETER:ITEM1:
CONDITION IN

:HISTory:PARameter:ITEM<x>:SOURce
Function Sets or queries the specified parameter's source trace for history-waveform parameter searches.
Syntax :HISTory:PARameter:ITEM<x>:
SOURce {<NRf>}
:HISTory:PARameter:ITEM<x>:SOURce?
<x> = 1 to 4
<NRf> = 1 to 16
Example :HISTORY:PARAMETER:ITEM1:SOURCE 1
:HISTORY:PARAMETER:ITEM1:SOURCE? ->
:HISTORY:PARAMETER:ITEM1:SOURCE 1

:HISTory:PARameter:ITEM<x>:TYPE?
Function Queries, for history-waveform parameter searches, the specified parameter's automatically measured item and upper and lower limits.
Syntax :HISTory:PARameter:ITEM<x>:TYPE?
<x> = 1 to 4

:HISTory:PARameter:ITEM<x>:TYPE:<Parameter>

Function Sets or queries the specified parameter's upper and lower limits for history-waveform parameter searches.

Syntax
```
:HISTory:PARameter:ITEM<x>:
TYPE:<Parameter>
{<{Voltage|DONTcare}>,
{<{Voltage|DONTcare}>,
<{Current|DONTcare}>,
<{Current|DONTcare}>|
<{Time|DONTcare}>,
<{Time|DONTcare}>|
<{Frequency|DONTcare}>,<{Frequency|
DONTcare}>|<{<NRf>|DONTcare}>,
<{<NRf>|DONTcare}>}
<x> = 1 to 16
<Parameter> = {AMPLitude|AVERage|
AVGFreq|AVGPeriod|BWIDth1|BWIDth2|
DELay|DUTYcycle|FALL|FREQuency|HIGH|
LOW|MAXimum|MIDDle|MINimum|
NOVershoot|NWIDth|PERiod|PNUMber|
POVershoot|PTOPeak|PWIDth|RISE|RMS|
SDEViation|TY1Integ|TY2Integ}
```

:HISTory:PARameter:ITEM<x>:VALue?

Function Queries the specified parameter's measured value for history-waveform parameter searches.

Syntax
```
:HISTory:PARameter:ITEM<x>:TYPE:
VALue?
<x> = 1 to 4
```

Example
```
:HISTORY:PARAMETER:ITEM:VALUE? ->
:HISTORY:PARAMETER:ITEM1:
VALUE 150.25000E+00
```

:HISTory:PARameter:LOGic

Function Sets or queries the logic to apply to history waveform searches.

Syntax
```
:HISTory:PARameter:LOGic {AND|OR}
<x> = 1 to 4
```

Example
```
:HISTORY:PARAMETER:LOGIC AND
:HISTORY:PARAMETER:LOGIC? ->
:HISTORY:PARAMETER:LOGIC AND
```

:HISTory:PARameter:TRANge

Function Sets or queries the determination range of history waveform parameter searches.

Syntax
```
:HISTory:PARameter:
TRANge {<NRf>,<NRf>}
:HISTory:PARameter:TRANge?
<NRf> = -5 to 5 div
```

Example
```
:HISTORY:PARAMETER:TRANGE -5,5
:HISTORY:PARAMETER:TRANGE? ->
:HISTORY:PARAMETER:TRANGE -5.0000000
00000,5.000000000000
```

:HISTory:RECord

Function Sets or queries the source record.

Syntax
```
:HISTory:RECord {<NRf>|MINimum}
:HISTory:RECord?
<NRf> = 0 to -4999
Specify "MINimum" to specify the
minimum record number.
```

Example
```
:HISTORY:RECORD -1
:HISTORY:RECORD? -> :HISTORY:
RECORD -1
```

:HISTory:RECord? MINimum

Function Queries the minimum record number.

Example
```
:HISTORY:RECORD? MINimum
:HISTORY:RECORD -4
```

Description This command returns the record number of the oldest data in the history function.
During measurements, this value is fixed to 0.
Use this command when the DL850/DL850V is not performing measurements.

:HISTory:SMODe

Function Sets or queries the history waveform search mode.

Syntax
```
:HISTory:SMODe {OFF|ZONE|PARameter}
:HISTory:SMODe?
```

Example
```
:HISTORY:SMODE ZONE
:HISTORY:SMODE? -> :HISTORY:
SMODE ZONE
```

:HISTory:DATE?

Function Queries the trigger date of the data at the specified record number.

Syntax
```
:HISTory:DATE? {<NRf>|MINimum}
<NRf> = 0 to -4999
```

Example
```
:HIST:DATE? -1 -> :HIST:
DATE" -0001 2010/02/02"
```

Description If you specify a record number that is lower than MINimum, this command will return "----------."

:HISTory:TIME?

Function Queries the trigger time of the data at the specified record number.

Syntax
```
:HISTory:TIME? {<NRf>|MINimum}
<NRf> = 0 to –4999
```

Example
```
:HIST:TIME? -1 -> :HIST:
TIME" -0001 10:20:30.04"
```

Description If you specify a record number that is lower than MINimum, this command will return "----------."

:HISTory:ZONE?

Function Queries all history-waveform zone-search settings.

Syntax
```
:HISTory:ZONE?
```

:HISTory:ZONE:EDIT<x>?

Function Queries all settings for the specified search zone.

Syntax
```
:HISTory:ZONE:EDIT<x>?
<x> = 1 to 4
```

:HISTory:ZONE:EDIT<x>:CONDition

Function Sets or queries the specified search zone's search condition.

Syntax
```
:HISTory:ZONE:EDIT<x>:
CONDition {OFF|IN|OUT}
:HISTory:ZONE:EDIT<x>:CONDition?
```

Example
```
:HISTORY:ZONE:EDIT1:CONDITION IN
:HISTORY:ZONE:EDIT1:CONDITION? ->
:HISTORY:ZONE:EDIT1:CONDITION IN
```

:HISTory:ZONE:EDIT<x>:SOURce

Function Sets or queries the specified search zone's source waveform.

Syntax
```
:HISTory:ZONE:EDIT<x>:
SOURce {<NRf>}
:HISTory:ZONE:EDIT<x>:SOURce?
<x> = 1 to 4
<NRf> = 1 to 16
```

Example
```
:HISTORY:ZONE:EDIT1:SOURCE 1
:HISTORY:ZONE:EDIT1:SOURCE? ->
:HISTORY:ZONE:EDIT1:SOURCE 1
```

:HISTory:ZONE:LOGic

Function Sets or queries the logic condition of history-waveform zone searches.

Syntax
```
:HISTory:ZONE:LOGic {AND|OR}
:HISTory:ZONE:LOGic?
```

Example
```
:HISTORY:ZONE:LOGIC AND
:HISTORY:ZONE:LOGIC? -> :HISTORY
:ZONE:LOGIC AND
```

5.17 IMAGe Group

The commands in this group deal with saving screen capture data. You can perform the same operations and make the same settings and queries that you can by using the SHIFT+SAVE keys on the front panel.

:IMAGe?
Function Queries all screen capture data output settings.
Syntax `:IMAGe?`

:IMAGe:BACKground
Function Sets or queries the screen capture background (png).
Syntax `:IMAGe:BACKground {NORMal|TRANsparent}`
`:IMAGe:BACKground?`
Example `:IMAGE:BACKGROUND NORMAL`
`:IMAGE:BACKGROUND? -> :IMAGE:BACKGROUND NORMAL`

:IMAGe:COMMent
Function Sets or queries the screen comment.
Syntax `:IMAGe:COMMent {<String>}`
`:IMAGe:COMMent?`
`<String> = Up to 26 characters`
Example `:IMAGE:COMMENT "ABC"`
`:IMAGE:COMMENT? -? :IMAGE:COMMENT "ABC"`

:IMAGe:EXECute
Function Saves the screen capture data.
Syntax `:IMAGe:EXECute`
Example `:IMAGE:EXECUTE`

:IMAGe:FORMat
Function Sets or queries the screen capture output format.
Syntax `:IMAGe:FORMat {BMP|JPEG|PNG}`
`:IMAGe:FORMat?`
Example `:IMAGE:FORMAT PNG`
`:IMAGE:FORMAT? -> :IMAGE:FORMAT PNG`

:IMAGe:SAVE?
Function Queries all file output settings.
Syntax `:IMAGe:SAVE?`

:IMAGe:SAVE:ANAMing
Function Sets or queries the setting of the auto naming feature for saving files.
Syntax `:IMAGe:SAVE:ANAMing {DATE|NUMBering|OFF}`
`:IMAGe:SAVE:ANAMing?`
Example `:IMAGE:SAVE:ANAMING NUMBERING`
`:IMAGE:SAVE:ANAMING? -> :IMAGE:SAVE:ANAMING NUMBERING`

:IMAGe:SAVE:CDIRectory
Function Changes the output destination directory.
Syntax `:IMAGe:SAVE:CDIRectory {<String>}`
`<String> = Up to 16 characters`
Example `:IMAGW:SAVE:CDIRECTORY "20100318_000"`

:IMAGe:SAVE:DRIVe
Function Sets the output destination medium.
Syntax `:IMAGe:SAVE:DRIVe {IFMemory|HD,<NRf>|NETWork|SD|USB ,<NRf>,<NRf>}`
Example `:IMAGe:SAVE:DRIVe USB,0`
Description See the description of the :FILE[:DIRectory]: DRIVe command.

:IMAGe:SAVE:FRAMe
Function Sets or queries whether a white frame is attached to the saved screen capture's image.
Syntax `:IMAGe:SAVE:FRAMe {<Boolean>}`
`:IMAGe:SAVE:FRAMe?`
Example `:IMAGe:SAVE:FRAMe 1`
`:IMAGe:SAVE:FRAMe? -> :IMAGe:SAVE:FRAMe 1`

:IMAGe:SAVE:NAME
Function Sets or queries the name of the file that will be saved.
Syntax `:IMAGe:SAVE:NAME {<Filename>}`
`:IMAGe:SAVE:NAME?`
Example `:IMAGE:SAVE:NAME "ABC"`
`:IMAGe:SAVE:NAME? -> :IMAGe:SAVE:NAME`

:IMAGe:SAVE:PATH?
Function Queries the current directory.
Example `:IMAGE:SAVE:PATH? -> :IMAGE:SAVE:PATH "Path=HD,0"`

:IMAGe:SEND?
Function Queries the screen capture data.
Syntax `:IMAGe:SEND?`
Description The screen capture data is returned as block data.

:IMAGe:TONE
Function Sets or queries the color tone of the screen capture data that will be saved.
Description `:IMAGe:TONE {COLor|GRAY|OFF|REVerse}`
`:IMAGe:TONE?`
Example `:IMAGE:TONE REVERSE`
`:IMAGE:TONE? -> :IMAGE:TONE REVERSE`

5.18 INITialize Group

The commands in this group deal with initializing the DL850/DL850V settings. You can perform the same operations that you can by pressing the SETUP key on the front panel and then the Initialize soft key.

:INITialize:EXECute

Function Initializes the settings. This is an overlap command.

Syntax `:INITialize:EXECute`

Description
- The following settings cannot be reset.
  - Date and time settings
  - Communication settings
  - The language setting (English or Japanese)
  - Environment settings
- An error will occur if the DL850/DL850V receives this command while hard disk recording is being performed. When you want to execute this command, stop hard disk recording.
- History data is cleared.

:INITialize:UNDO

Function Undoes the setting initialization.

Syntax `:INITialize:UNDO`

Description The cleared history data will not be restored.

5.19 LSTart Group

The commands in this group deal with starting the log operation. These commands set the trigger mode to On Start and start waveform acquisition.

:LSTart (Log STart)

Function Starts waveform acquisition immediately.
Syntax `:LSTart`

:LSTart?

Function Starts waveform acquisition immediately, and waits for acquisition to complete.
Syntax `:LSTart?`
Example `:LSTart? -> 0`
Description When acquisition is complete, this command returns 0.

5.20 MATH Group

The commands in this group deal with computations. You can make the same settings and queries that you can by using the MATH key on the front panel.
If the selectable range of the time axis is "<NRf> = –5 to 5div," the selectable range varies depending on settings such as the record length. For details, see "Selectable Range of Cursor Positions" under "Notes about Cursor Measurement" in chapter 9 of IM DL850-01EN.

:MATH<x>?

Function Queries all computation settings.
Syntax
```
:MATH<x>?
<x> = 1 to 8
```

:MATH<x>:AVERage?

Function Queries all averaging computation settings.
Syntax
```
:MATH<x>:AVERage?
<x> = 1 to 8 (this command performs
the same no matter what value you
specify)
```
Description This command is valid on models with the /G2 option.

:MATH<x>:AVERage:CCOunt

Function Sets or queries the cycle count of cycle averaging.
Syntax
```
:MATH<x>:AVERage:CCOunt {<NRf>}
:MATH<x>:AVERage:CCOunt?
<x> = 1 to 8 (this command performs
the same no matter what value you
specify)
<NRf> = 10 to 1800
```
Example
```
:MATH1:AVERAGE:CCOUNT 720
:MATH1:AVERAGE:CCOUNT? -> :MATH1:
AVERAGE:CCOUNT 720
```
Description This command is valid on models with the /G2 option.

:MATH<x>:AVERage:COUNt

Function Sets or queries the average count of linear averaging.
Syntax
```
:MATH<x>:AVERage:COUNt {<NRf>}
:MATH<x>:AVERage:COUNt?
<x> = 1 to 8 (this command performs
the same no matter what value you
specify)
<NRf> = 2 to 128 (in 2n steps)
```
Example
```
:MATH1:AVERAGE:COUNT 16
:MATH1:AVERAGE:COUNT? -> :MATH1:
AVERAGE:COUNT 16
```
Description This command is valid on models with the /G2 option.

:MATH<x>:AVERage:EWEight

Function Sets or queries the attenuation constant of exponential averaging.
Syntax
```
:MATH<x>:AVERage:EWEight {<NRf>}
:MATH<x>:AVERage:EWEight?
<x> = 1 to 8 (this command performs
the same no matter what value you
specify)
<NRf> = 2 to 256 (in 2n steps)
```
Example
```
:MATH1:AVERAGE:EWEIGHT 16
:MATH1:AVERAGE:EWEIGHT? -> :MATH1:
AVERAGE:EWEIGHT 16
```
Description This command is valid on models with the /G2 option.

:MATH<x>:AVERage:MODE

Function Sets or queries the averaging mode.
Syntax
```
:MATH<x>:AVERage:MODE {CYCLe|
EXPonent|LINear|OFF|PEAK}
:MATH<x>:AVERage:MODE?
<x> = 1 to 8 (this command performs
the same no matter what value you
specify)
```
Example
```
:MATH1:AVERAGE:MODE EXPONENT
:MATH1:AVERAGE:MODE? -> :MATH1:
AVERAGE:MODE EXPONENT
```
Description This command is valid on models with the /G2 option.

:MATH<x>:AVERage:TYPE

Function Sets or queries the averaging domain.
Syntax
```
:MATH<x>:AVERage:TYPE {FREQuency|
TIME}
:MATH<x>:AVERage:TYPE?
<x> = 1 to 8 (this command performs
the same no matter what value you
specify)
```
Example
```
:MATH1:AVERAGE:TYPE FREQUENCY
:MATH1:AVERAGE:TYPE? -> :MATH1:
AVERAGE:TYPE FREQUENCY
```
Description This command is valid on models with the /G2 option.

:MATH<x>:BINary?

Function Queries all binary computation settings.
Syntax
```
:MATH<x>:BINary?
```

:MATH<x>:BINary:CHANnel<x>:SCHannel<x>[:THReshold]

Function Sets or queries the threshold level of the specified channel for binary computations.

Syntax
```
:MATH<x>:BINary:{CHANnel<x>|MATH<x>}
[:THReshold] {<Voltage>,<Voltage>|
<Current>,<Current>|<NRf>,<NRf>}
:MATH<x>:BINary:{CHANnel<x>|
MATH<x>}[:THReshold]?
:MATH<x>:BINary:CHANnel<x>:
SCHannel<x>[:THReshold] {<Voltage>,
<Voltage>|<Current>,<Current>|
<NRf>,<NRf>}
:MATH<x>:BINary:CHANnel<x>:
SCHannel<x>[:THReshold]?
MATH<x>'s <x> = 1 to 8
CHANnel<x>'s <x> = 1 to 16
SCHannel<x>'s <x> = 1 to 16 (for
modules that have sub channels)
<Voltage>, <Current>, and <NRf> =
The selectable range varies depending
on settings such as the range and
offset. For details, see the DL850/
DL850V User's Manual.
For settings other than MATH<x>,
<NRf> = -10 to 10 div (or equivalent
values)
For MATH<x>, <NRf> = -5 to 5 (div;
in 0.01 div steps)
```

Example
```
:MATH1:BINARY:CHANNEL1:THRESHOLD 1,0
:MATH1:BINARY:CHANNEL1:THRESHOLD? ->
:MATH1:BINARY:CHANNEL1:
THRESHOLD 1.000000E+00,0.000000E+00
```

:MATH<x>:CONStant<x>

Function Sets or queries a constant for user-defined computation.

Syntax
```
:MATH<x>:CONStant<x> {<NRf>}
:MATH<x>:CONStant<x>?
MATH<x>'s <x> = 1 to 8
CONStant<x>'s <x> = 1 to 8
<NRf> = -9.9999E+30 to +9.9999E+30
```

Example
```
:MATH1:CONSTANT1 1
:MATH1:CONSTANT1? -> :MATH1:
CONSTANT1 1
```

Description This command is valid on models with the /G2 option.

:MATH<x>:DEFine

Function Sets or queries an expression for user-defined computation.

Syntax
```
:MATH<x>:DEFine {<String>}
:MATH<x>:DEFine?
MATH<x>'s <x> = 1 to 8
<String> = Up to 80 characters
```

Example
```
:MATH1:DEFINE "PS(C1)"
:MATH1:DEFINE? -> :MATH1:
DEFINE "PS(C1)"
```

Description If OPERation is set to a value other than USERdefine, this query will return the OPERation setting.
This command is valid on models with the /G2 option.

:MATH<x>:DISPlay

Function Sets or queries whether computations will be performed.

Syntax
```
:MATH<x>:DISPlay {<Boolean>}
:MATH<x>:DISPlay?
MATH<x>'s <x> = 1 to 8
```

Example
```
:MATH1:DISPLAY ON
:MATH1:DISPLAY? -> :MATH1:DISPLAY 1
```

:MATH<x>:ESHift:CHANnel<x>:SCHannel<x>[:COUNt]

Function Sets or queries the amount of phase shift in the Shift computation when an external clock is used.

Syntax
```
:MATH<x>:ESHift:{CHANnel<x>|
MATH<x>}[:COUNt] {<NRf>}
:MATH<x>:ESHift:CHANnel<x>:
SCHannel<x>} [:COUNt] {<NRf>}
:MATH<x>:ESHift:{CHANnel<x>|
MATH<x>}[:COUNt]?
:MATH<x>:ESHift:CHANnel<x>:
SCHannel<x>} [:COUNt]?
MATH<x>'s <x> = 1 to 8
<NRf> = ±(record length/2). The
resolution is 1.
```

Example
```
:MATH1:ESHIFT:CHANNEL1:COUNT 10
:MATH1:ESHIFT:CHANNEL1:COUNT? ->
:MATH1:ESHIFT:CHANNEL1:COUNT 10
```

:MATH<x>:EXPonential?

Function Queries all exponential window settings.

Syntax
```
:MATH<x>:EXPonential?
<x> = 1 to 8 (this command performs
the same no matter what value you
specify)
```

Description This command is valid on models with the /G2 option.

:MATH<x>:EXPonential:DRATe

Function Sets or queries the exponential window's damping rate.

Syntax
```
:MATH<x>:EXPonential:DRATe {<NRf>}
:MATH<x>:EXPonential:DRATe?
<x> = 1 to 8 (this command performs the same no matter what value you specify)
<NRf> = 1 to 100
```

Example
```
:MATH1:EXPONENTIAL:DRATE 100
:MATH1:EXPONENTIAL:DRATE? -> :MATH1:EXPONENTIAL:DRATE 100
```

Description This command is valid on models with the /G2 option.

:MATH<x>:EXPonential:FORCe<x>

Function Sets or queries the exponential window's FORCe1 or FORCe2 value.

Syntax
```
:MATH<x>:EXPonential:FORCe<x> {<NRf>}
:MATH<x>:EXPonential:FORCe<x>?
MATH<x> = 1 to 8 (this command performs the same no matter what value you specify)
FORCe<x> = 1, 2
<NRf> = 1 to 100
```

Example
```
:MATH1:EXPONENTIAL:FORCE1 100
:MATH1:EXPONENTIAL:FORCE1? -> :MATH1:EXPONENTIAL:FORCE1 100
```

Description This command is valid on models with the /G2 option.

:MATH<x>:FFT?

Function Queries all FFT computation settings.

Syntax
```
:MATH<x>:FFT?
<x> = 1 to 8 (this command performs the same no matter what value you specify)
```

:MATH<x>:FFT:POINt

Function Sets or queries the number of FFT points.

Syntax
```
:MATH<x>:FFT:POINt {<NRf>}
:MATH<x>:FFT:POINt?
MATH<x> = 1 to 8 (this command performs the same no matter what value you specify)
<NRf> = 1000, 2000, 5000, 10000, 20000, 50000, 100000
```

Example
```
:MATH1:FFT:POINT 1000
:MATH1:FFT:POINT? -> :MATH1:FFT:POINT 1000
```

:MATH<x>:FFT:WINDow

Function Sets or queries the time window for FFT computations.

Syntax
```
:MATH<x>:FFT:WINDow {HANNing|RECTangle|FLATtop|EXPonential|HAMMing}
:MATH<x>:FFT:WINDow?
MATH<x> = 1 to 8 (this command performs the same no matter what value you specify)
```

Example
```
:MATH1:FFT:WINDOW HANNING
:MATH1:FFT:WINDOW? -> :MATH1:FFT:WINDOW HANNING
```

Description The EXPonential parameter is valid on models with the /G2 option.

:MATH<x>:FILTer<x>?

Function Queries all digital filter settings.

Syntax
```
:MATH<x>:FILTer<x>?
MATH<x> = 1 to 8 (this command performs the same no matter what value you specify)
FILTer<x> = 1, 2
```

Description This command is valid on models with the /G2 option.

:MATH<x>:FILTer<x>:BAND

Function Sets or queries a digital filter band.

Syntax
```
:MATH<x>:FILTer<x>:BAND {BPASs|HPASs|LPASs}
:MATH<x>:FILTer<x>:BAND?
MATH<x> = 1 to 8 (this command performs the same no matter what value you specify)
FILTer<x> = 1, 2
```

Example
```
:MATH1:FILTER1:BAND LPASS
:MATH1:FILTER1:BAND? -> :MATH1:FILTER1:BAND LPASS
```

Description You can only specify the GAUSs parameter when the band is set to LPASs.
This command is valid on models with the /G2 option.

:MATH<x>:FILTer<x>:CUToff<x>

Function Sets or queries a cutoff frequency of a digital filter.

Syntax
```
:MATH<x>:FILTer<x>:CUToff<x> {<NRf>}
:MATH<x>:FILTer<x>:CUToff<x>?
MATH<x> = 1 to 8 (this command performs the same no matter what value you specify)
FILTer<x> = 1, 2
CUToff<x> = 1, 2
<NRf> = 2 to 30% (in 0.2% steps)
```

Example
```
:MATH1:FILTER1:CUTOFF1 10.0
:MATH1:FILTER1:CUTOFF1? -> :MATH1:FILTER1:CUTOFF1 10.0
```

Description This command is valid on models with the /G2 option.

:MATH<x>:FILTer<x>:TYPE

Function Sets or queries the type of a digital filter.

Syntax
```
:MATH<x>:FILTer<x>:TYPE {GAUSs|IIR|SHARp}
:MATH<x>:FILTer<x>:TYPE?
MATH<x> = 1 to 8 (this command performs the same no matter what value you specify)
FILTer<x> = 1, 2
```

Example
```
:MATH1:FILTER1:TYPE GAUSS
:MATH1:FILTER1:TYPE? -> :MATH1:FILTER1:TYPE GAUSS
```

Description This command is valid on models with the /G2 option.

:MATH<x>:LABel

Function Sets or queries a computed waveform label.

Syntax
```
:MATH<x>:LABel {<String>}
:MATH<x>:LABel?
MATH<x> = 1 to 8 (this command
performs the same no matter what
value you specify)
<String> = Up to 16 characters
```

Example
```
:MATH1:LABEL "ABC"
:MATH1:LABEL? -> :MATH1:LABEL "ABC"
```

:MATH<x>:MODE

Function Sets or queries whether the DL850/DL850V is in computation mode.

Syntax
```
:MATH<x>:MODE {<Boolean>}
:MATH<x>:MODE?
MATH<x> = 1 to 8 (this command
performs the same no matter what
value you specify)
```

Example
```
:MATH1:MODE ON
:MATH1:MODE? -> :MATH1:MODE 1
```

:MATH<x>:MREFerence

Function Sets or queries the computation range.

Syntax
```
:MATH<x>:MREFerence {<NRf>,<NRf>}
:MATH<x>:MREFerence?
MATH<x> = 1 to 8 (this command
performs the same no matter what
value you specify)
<NRf> = −5 to 5 div
```

Example
```
:MATH1:MREFERENCE -5,5
:MATH1:MREFERENCE? -> :MATH1
:MREFERENCE -5.000000000000,
5.000000000000
```

:MATH<x>:OPERation

Function Sets or queries the computation type.

Syntax
```
:MATH<x>:OPERation {PLUS|MINus|
MULTiple|BINary|DIVIde|SHIFt|PS|OFF|
USERdefine}
:MATH<x>:OPERation?
MATH<x>'s <x> = 1 to 8
```

Example
```
:MATH1:OPERATION PS
:MATH1:OPERATION? -> :MATH1:
OPERATION PS
```

Description The USERdefine parameter is valid on models with the /G2 option.

:MATH<x>:SOURce<x>

Function Sets or queries a sub channel's computation type.

Syntax
```
:MATH<x>:SOURce<x> {<NRf>[,<NRf>]|
MATH<x>}
:MATH<x>:SOURce<x>?
MATH<x>'s <x> = 1 to 8
SOURce<x>'s <x> = 1, 2
```

Description You cannot use this command when no modules are installed.

:MATH<x>:SCALe?

Function Queries all scaling settings.

Syntax
```
:MATH<x>:SCALe?
MATH<x> = 1 to 8
```

:MATH<x>:SCALe:MODE

Function Sets or queries a scale mode.

Syntax
```
:MATH<x>:SCALe:MODE {AUTO|MANual}
:MATH<x>:SCALe:MODE?
MATH<x> = 1 to 8
```

Example
```
:MATH1:SCALE:MODE AUTO
:MATH1:SCALE:MODE? -> :MATH1:SCALE
:MODE AUTO
```

:MATH<x>:SCALe:VALue

Function Sets or queries a set of upper and lower limits for manual scaling.

Syntax
```
:MATH<x>:SCALe:VALue {<NRf>,<NRf>}
:MATH<x>:SCALe:VALue?
MATH<x> = 1 to 8
<NRf> = −9.9999E+30 to +9.9999E+30
```

Example
```
:MATH1:SCALE:VALUE 1,-1
:MATH1:SCALE:VALUE? -> :MATH1:SCALE:
VALUE 1.00000E+00,-1.00000E+00
```

:MATH<x>:SHIFt:CHANnel<x>: SCHannel<x>[:TIME]

Function Sets or queries the amount of phase shift in the Shift computation when the internal clock is used.

Syntax
```
:MATH<x>:SHIFt:{CHANnel<x>|MATH<x>}
[:TIME] {<Time>}
:MATH<x>:SHIFt:CHANnel<x>:
SCHannel<x>[:TIME] {<Time>}
:MATH<x>:SHIFt:{CHANnel<x>|MATH<x>}
[:TIME]?
:MATH<x>:SHIFt:CHANnel<x>:
SCHannel<x>[:TIME]?
MATH<x>'s <x> = 1 to 8
<Time> = 0 s to 2592000 s (720
hours) ± (T/Div × 5)
Step: 1 / sample rate
```

Example
```
:MATH1:SHIFT:CHANNEL1:TIME -0.001
:MATH1:SHIFT:CHANNEL1:TIME? ->
:MATH1:SHIFT:CHANNEL1:
TIME -1.000000000E-03
```

:MATH<x>:UNIT

Function Sets or queries a unit string that is attached to computation results.

Syntax
```
:MATH<x>:UNIT {<String>}
:MATH<x>:UNIT?
MATH<x>'s <x> = 1 to 8
<String> = Up to 4 characters
```

Example
```
:MATH1:UNIT "UU"
:MATH1:UNIT? -> :MATH1:UNIT "UU"
```

5.21 MEASure Group

The commands in this group deal with the automated measurement of waveform parameters. You can make the same settings and queries (of settings and measured values) that you can by using the MEASURE key on the front panel. If the selectable range of the time axis is "<NRf> = –5 to 5div," the selectable range varies depending on settings such as the record length. For details, see "Selectable Range of Cursor Positions" under "Notes about Cursor Measurement" in chapter 9 of IM DL850-01EN.

:MEASure?

Function Queries all the settings for automated measurement of waveform parameters.

Syntax `:MEASure?`

:MEASure:CHANnel<x>:SCHannel<x>?

Function Sets or queries whether all the waveform parameters of the specified channel are ON or OFF.

Syntax
```
:MEASure:{CHANnel<x>|MATH<x>}?
:MEASure:CHANnel<x>:SCHannel<x>?
<x> = 1 to 16
<x> = 1 to 16 (sub channel)
MATH<x>'s <x> = 1 to 8
```

Description For channels that have sub channels, use the :MEAS:CHAN:SCH? command.

:MEASure:CHANnel<x>:SCHannel<x>:ALL

Function Sets all the measurement items of the specified channel to ON or OFF.

Syntax
```
:MEASure:{CHANnel<x>|MATH<x>}:
ALL {<Boolean>}
:MEASure:CHANnel<x>:SCHannel<x>:
ALL {<Boolean>}
<x> = 1 to 16
<x> = 1 to 16 (sub channel)
MATH<x>'s <x> = 1 to 8
```

Example
```
:MEASURE:CHANNEL1:ALL ON
:MEASURE:CHANNEL7:SCHANNEL1:ALL ON
```

Description For channels that have sub channels, use the :MEAS:CHAN:SCH:ALL command.

:MEASure:CHANnel<x>:SCHannel<x>:CDEStination (CopyDestination)

Function Sets or queries whether the copy destination channel is ON or OFF for when parameter measurement items are copied between channels.

Syntax
```
:MEASure:{CHANnel<x>|MATH<x>}:
CDEStination {<Boolean>}
:MEASure:CHANnel<x>:SCHannel<x>:
CDEStination
<x> = 1 to 16
<x> = 1 to 16 (sub channel)
MATH<x>'s <x> = 1 to 8
```

Example
```
:MEASURE:CHANNEL2:CDESTINATION ON
:MEASURE:CHANNEL2:CDESTINATION? ->
:MEASURE:CHANNEL2:CDESTINATION 1
:MEASURE:CHANNEL7:SCHANNEL1:
CDESTINATION ON
:MEASURE:CHANNEL7:SCHANNEL1:
CDESTINATION? -> :MEASURE:CHANNEL7:
SCHANNEL1:CDESTINATION 1
```

:MEASure:CHANnel<x>:SCHannel<x>:COPY

Function Copies all the measurement item ON/OFF settings from one specified channel to another specified channel.

Syntax
```
:MEASure:{CHANnel<x>|MATH<x>}:COPY
:MEASure:CHANnel<x>:SCHannel<x>:COPY
<x> = 1 to 16
<x> = 1 to 16 (sub channel)
MATH<x>'s <x> = 1 to 8
```

Example `:MEASURE:CHANNEL1:COPY`

:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DELay?

Function Queries all delay settings.

Syntax
```
:MEASure:{CHANnel<x>[:SCHannel<x>]|
MATH<x>}:DELay?
```

:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DELay:COUNt?

Function Queries the statistics count of the delay between channels.

Syntax
```
:MEASure:{CHANnel<x>[:SCHannel<x>]|
MATH<x>}:DELay:COUNt?
```

Example
```
:MEASURE:CHANNEL1:DELAY:COUNT? ->
:MEASURE:CHANNEL1:DELAY:COUNT 3
```

:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DELay:{MAXimum|MEAN|MINimum|SDEViation}?

Function Queries a statistic of the delay between channels.

Syntax
```
:MEASure:{CHANnel<x>[:SCHannel<x>|
MATH<x>}:DELay:{MAXimum|MEAN|MINimum|
SDEViation}?
CHANnel<x>'s <x> = 1 to 16
MATH<x>'s <x> = 1 to 8
```

Example
```
:MEASURE:CHANNEL1:DELAY:MEAN? ->
:MEASURE:CHANNEL1:DELAY:
MEAN 6.6666667E-06
```

:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DELay:MEASure?

Function Queries all the settings for a source waveform for measuring the delay between channels of a waveform.

Syntax
```
:MEASure:{CHANnel<x>[:SCHannel<x>|
MATH<x>}:DELay:MEASure?
CHANnel<x>'s <x> = 1 to 16
MATH<x>'s <x> = 1 to 8
```

:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DELay:MEASure:COUNt

Function Sets or queries the number of source waveforms at which delay between channels will be measured for a waveform.

Syntax
```
:MEASure:{CHANnel<x>[:SCHannel<x>]|
MATH<x>}:DELay:MEASure:COUNt {<NRf>}
:MEASure:{CHANnel<x>[:SCHannel<x>]|
MATH<x>}:DELay:MEASure:COUNt?
<NRf> = 1 to 9
```

Example
```
:MEASURE:CHANNEL1:DELAY:MEASURE:
COUNT 2
:MEASURE:CHANNEL1:DELAY:MEASURE:
COUNT? -> :MEASURE:CHANNEL1:DELAY:
MEASURE:COUNT 2
```

:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DELay:MEASure:SLOPe

Function Sets or queries a source waveform slope that will be used to measure delay between channels for a waveform.

Syntax
```
:MEASure:{CHANnel<x>[:SCHannel<x>]|
MATH<x>}:DELay:MEASure:
SLOPe {RISE|FALL}
:MEASure:{CHANnel<x>[:SCHannel<x>]|
MATH<x>}:DELay:MEASure:SLOPe?
```

Example
```
:MEASURE:CHANNEL1:DELAY:MEASURE:
SLOPE RISE
:MEASURE:CHANNEL1:DELAY:MEASURE:
SLOPE? -> :MEASURE:CHANNEL1:DELAY:
MEASURE:SLOPE RISE
```

:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DELay:REFerence?

Function Queries all reference waveform settings used to measure the delay between channels for the specified waveform.

Syntax
```
:MEASure:{CHANnel<x>[:SCHannel<x>]|
MATH<x>}:DELay:REFerence?
```

:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DELay:REFerence:COUNt

Function Sets or queries the edge detection count of the reference waveform used to measure the delay between channels for the specified waveform.

Syntax
```
:MEASure:{CHANnel<x>[:SCHannel<x>]|
MATH<x>}:DELay:REFerence:
COUNt {<NRf>}
:MEASure:{CHANnel<x>[:SCHannel<x>]|
MATH<x>}:DELay:REFerence:
COUNt {<NRf>}?
<NRf> = 1 to 9
```

Example
```
:MEASURE:CHANNEL1:DELAY:REFERENCE:
COUNT 1
:MEASURE:CHANNEL1:DELAY:REFERENCE:
COUNT? -> :MEASURE:CHANNEL1:DELAY:
REFERENCE:COUNT 1
```

:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DELay:REFerence:SLOPe

Function Sets or queries the edge detection slope of the reference waveform used to measure the delay between channels for the specified waveform.

Syntax
```
:MEASure:{CHANnel<x>[:SCHannel<x>]|
MATH<x>}:DELay:REFerence:
SLOPe {RISE|FALL}
:MEASure:{CHANnel<x>[:SCHannel<x>]|
MATH<x>}:DELay:REFerence:SLOPe?
```

Example
```
:MEASURE:CHANNEL1:DELAY:REFERENCE:
SLOPE RISE
:MEASURE:CHANNEL1:DELAY:REFERENCE:
SLOPE? -> :MEASURE:CHANNEL1:DELAY:
REFERENCE:SLOPE RISE
```

:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DELay:REFerence:SOURce

Function Sets or queries whether to set the reference point for measuring the delay between channels for a waveform to a trigger point or to a waveform.

Syntax
```
:MEASure:{CHANnel<x>[:SCHannel<x>]|
MATH<x>}:DELay:REFerence:
SOURce {TRACe|TRIGger}
:MEASure:{CHANnel<x>[:SCHannel<x>]|
MATH<x>}:DELay:REFerence:SOURce?
```

Example
```
:MEASURE:CHANNEL1:DELAY:REFERENCE:
SOURCE TRACE
:MEASURE:CHANNEL1:DELAY:REFERENCE:
SOURCE? -> :MEASURE:CHANNEL1:DELAY:
REFERENCE:SOURCE TRACE
```

:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DELay:REFerence:TRACe

Function Sets or queries the reference waveform trace used to measure the delay between channels for a waveform.

Syntax
```
:MEASure:{CHANnel<x>[:SCHannel<x>]|
MATH<x>}:DELay:REFerence:
TRACe {<NRf>[,<NRf>]|MATH<x>}
:MEASure:{CHANnel<x>[:SCHannel<x>]|
MATH<x>}:DELay:REFerence:TRACe?
```

Example
```
:MEASURE:CHANNEL1:DELAY:REFERENCE:
TRACE 1
:MEASURE:CHANNEL1:DELAY:REFERENCE:
TRACE? -> :MEASURE:CHANNEL1:DELAY:
REFERENCE:TRACE 1
```

:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DELay:STATe

Function Sets or queries the display format of the delay parameters of the specified channel.

Syntax
```
:MEASure:{CHANnel<x>[:SCHannel<x>]|
MATH<x>}:DELay:STATe {TIME|OFF|
DEGRee}
:MEASure:{CHANnel<x>[:SCHannel<x>]|
MATH<x>}:DELay:STATe?
```

Example
```
:MEASURE:CHANNEL1:DELAY:STATE TIME
:MEASURE:CHANNEL1:DELAY:STATE? ->
:MEASURE:CHANNEL1:DELAY:STATE TIME
```

:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DELay:VALue?

Function Queries a measured delay value of the specified waveform's parameter.

Syntax
```
:MEASure:{CHANnel<x>[:SCHannel<x>]|
MATH<x>}:DELay:VALue? {<NRf>}
```

Example
```
:MEASURE:CHANNEL1:DELAY:VALUE? ->
:MEASURE:CHANNEL1:DELAY:
VALUE 0.0000000E+00
```

Description
- This command returns "NAN" if the value cannot be measured.
- <NRf> is used to specify which iteration of statistical processing to query the parameter value from. This command returns "NAN" if the specified value does not exist.
- <NRf> can be omitted. If it is omitted, the measured values of the newest waveform in history memory are queried. If you include <NRf>, the measured value of the waveform <NRf> times before the newest history waveform will be queried.

:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DPRoximal?

Function Queries all distal, mesial, and proximal settings.

Syntax
```
:MEASURE:CHANNEL1:DPROXIMAL?
```

:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DPRoximal:MODE

Function Sets or queries the distal, mesial, and proximal point mode setting.

Syntax
```
:MEASure:CHANnel<x>:DPRoximal:
MODE {PERCent|UNIT}
:MEASure:CHANnel<x>:DPRoximal:MODE?
```

Example
```
:MEASURE:CHANNEL1:DPROXIMAL:
MODE PERCENT
:MEASURE:CHANNEL1:DPROXIMAL:MODE? ->
:MEASURE:CHANNEL1:DPROXIMAL:
MODE PERCENT
```

:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DPRoximal:PERCent

Function Sets or queries the distal, mesial, and proximal points as percentages.

Syntax
```
:MEASure:{CHANnel<x>[:SCHannel<x>]|
MATH<x>}:DPRoximal:
PERCent {<NRf>,<NRf>,<NRf>}
:MEASure:{CHANnel<x>[:SCHannel<x>]|
MATH<x>}:DPRoximal:PERCent?
<NRf> = 0 to 100 (%; in 0.1% steps)
CHANnel<x>'s <x> = 1 to 16
SCHannel<x>'s <x> = 1 to 16
MATH<x>'s <x> = 1 to 8
The order is <Proximal><Mesial><Distal>.
```

Example
```
:MEASURE:CHANNEL1:DPROXIMAL:
PERCENT 20,50,80
:MEASURE:CHANNEL1:DPROXIMAL:PERCENT?
-> :MEASURE:CHANNEL1:DPROXIMAL:
PERCENT 20.0,50.0,80.0
```

:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:DPRoximal:UNIT

Function Sets or queries the distal, mesial, and proximal points as units.

Syntax
```
:MEASure:{CHANnel<x>[:SCHannel<x>]|
MATH<x>}:DPRoximal:UNIT {<Voltage>,
<Voltage>,<Voltage>|<Current>,<Curre
nt>,<Current>|<NRf>,<NRf>,<NRf>}
:MEASure:{CHANnel<x>[:SCHannel<x>]|
MATH<x>}:DPRoximal:UNIT?
<Voltage>, <Current>, and <NRf> = The
selectable range varies depending
on settings such as the range and
offset. For details, see the DL850/
DL850V User's Manual.
CHANnel<x>'s <x> = 1 to 16
SCHannel<x>'s <x> = 1 to 16
MATH<x>'s <x> = 1 to 8
The order is <Proximal><Mesial><Distal>.
Selectable range of the
MEAS:{CHAN<x>}:DPR:UNIT command
• For <Voltage>,<Voltage>,
  <Voltage>, the selectable range
  is ±the measurement range, and
  the resolution is the same as the
  resolution of the zoom method (DIV/
  SPAN) that has been set by the V
  Scale of the vertical axis.
For Math channels: ±5 div
```

Example
```
:MEASURE:CHANNEL1:DPROXIMAL:
UNIT -10,0,10
:MEASURE:CHANNEL1:DPROXIMAL:UNIT? ->
:MEASURE:CHANNEL1:DPROXIMAL:
UNIT -10.0000E+00,0.00000E+00,
10.0000E+00
```

Description You cannot set the distal, mesial, and proximal points using units on frequency modules. Depending on the settings that were in use before you sent this command, the values may not be set according to the parameters.

:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:METHod

Function Sets or queries the modes of a set of high and low points (rising-time and falling-time measurement references).

Syntax
```
:MEASure:{CHANnel<x>[:SCHannel<x>]|
MATH<x>}:METHod {AUTO|MAXMin}
:MEASure:{CHANnel<x>[:SCHannel<x>]|
MATH<x>}:METHod?
```

Example
```
:MEASURE:CHANNEL1:METHOD AUTO
:MEASURE:CHANNEL1:METHOD? ->
:MEASURE:CHANNEL1:METHOD AUTO
```

:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:<Parameter>?

Function Queries the setting of a waveform parameter (measurement item).

Syntax
```
:MEASure:{CHANnel<x>[:SCHannel<x>]|
MATH<x>}:<Parameter>?
CHANnel<x>'s <x> = 1 to 16
SCHannel<x>'s <x> = 1 to 16
MATH<x>'s <x> = 1 to 8
<Parameter> = {AMPLitude|AVERage|
AVGFreq|AVGPeriod|BWIDth1|BWIDth2|
DUTYcycle|FALL|FREQuency|HIGH|LOW|
MAXimum|MIDDle|MINimum|NOVershoot|
NWIDth|PERiod|PNUMber|POVershoot|
PTOPeak|PWIDth|RISE|RMS|SDEViation|
TY1Integ|TY2Integ}
```

Example
```
:MEASURE:CHANNEL1:PTOPEAK? ->
:MEASURE:CHANNEL1:PTOPEAK:STATE 0
```

Description This command returns the same response as the :MEAS:CHAN:<Parameter>:STAT? command.

:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:<Parameter>:COUNt?

Function Queries the count of measured values for cyclic statistical processing of a waveform parameter.

Syntax
```
:MEASure:{CHANnel<x>[:SCHannel<x>]|
MATH<x>}:<Parameter>:COUNt?
<Parameter> = {AMPLitude|AVERage|
AVGFreq|AVGPeriod|BWIDth1|BWIDth2|
DUTYcycle|FALL|FREQuency|HIGH|LOW|
MAXimum|MIDDle|MINimum|NOVershoot|
NWIDth|PERiod|PNUMber|POVershoot|
PTOPeak|PWIDth|RISE|RMS|SDEViation|
TY1Integ|TY2Integ}
```

Example
```
:MEASURE:CHANNEL1:PTOPEAK:COUNT? ->
:MEASURE:CHANNEL1:PTOPEAK:COUNT 3
```

:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:<Parameter>:{MAXimum|MEAN|MINimum|SDEViation}?

Function Queries a cyclic statistical processing value of a waveform parameter.

Syntax
```
:MEASure:{CHANnel<x>[:SCHannel<x>]|
MATH<x>}:<Parameter>:{MAXimum|MEAN|
MINimum|SDEViation}?
<Parameter> = {AMPLitude|AVERage|
AVGFreq|AVGPeriod|BWIDth1|BWIDth2|
DUTYcycle|FALL|FREQuency|HIGH|LOW|
MAXimum|MIDDle|MINimum|NOVershoot|
NWIDth|PERiod|PNUMber|POVershoot|
PTOPeak|PWIDth|RISE|RMS|SDEViation|
TY1Integ|TY2Integ}
```

Example
```
:MEASURE:CHANNEL1:PTOPEAK:MAXIMUM?
-> :MEASURE:CHANNEL1:PTOPEAK:
MAXIMUM 30.633333E+00
```

Description This command returns “NAN” for any statistics that cannot be processed.

:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:<Parameter>STATe

Function Sets or queries whether the specified waveform’s waveform parameter (measurement item) is ON or OFF.

Syntax
```
:MEASure:{CHANnel<x>[:SCHannel<x>]|
MATH<x>}:<Parameter>STATe
<Parameter> = {AMPLitude|AVERage|
AVGFreq|AVGPeriod|BWIDth1|BWIDth2|
DUTYcycle|FALL|FREQuency|HIGH|LOW|
MAXimum|MIDDle|MINimum|NOVershoot|
NWIDth|PERiod|PNUMber|POVershoot|
PTOPeak|PWIDth|RISE|RMS|SDEViation|
TY1Integ|TY2Integ}
```

Example
```
:MEASURE:CHANNEL1:PTOPEAK:STATE ON
:MEASURE:CHANNEL1:PTOPEAK:STATE? ->
:MEASURE:CHANNEL1:PTOPEAK:STATE 1
```

:MEASure:{CHANnel<x>[:SCHannel<x>]|MATH<x>}:<Parameter>VALue?

Function Queries the measured value of a waveform parameter.

Syntax
```
:MEASure:{CHANnel<x>[:SCHannel<x>]|
MATH<x>}:<Parameter>VALue? {<NRf>}
CHANnel<x>'s <x> = 1 to 16
<NRf> = 1 to 48000
<Parameter> = {AMPLitude|AVERage|
AVGFreq|AVGPeriod|BWIDth1|BWIDth2|
DUTYcycle|FALL|FREQuency|HIGH|LOW|
MAXimum|MIDDle|MINimum|NOVershoot|
NWIDth|PERiod|PNUMber|POVershoot|
PTOPeak|PWIDth|RISE|RMS|SDEViation|
TY1Integ|TY2Integ}
```

Example
```
:MEASURE:CHANNEL1:PTOPEAK:VALUE? ->
:MEASURE:CHANNEL1:PTOPEAK:
VALUE 30.516667E+00
```

Description This command returns “NAN” if the value cannot be measured.
Parameters cannot be measured if the measurement has not been performed or if the measurement was performed, but computations were not performed because the specified range exceeded 10 Mpoint.
The <NRf> at the end is used to specify which iteration of statistical processing to query the parameter value from. This command returns “NAN” (Not A Number) if the specified value does not exist.

- For non-cyclic statistical processing. <NRf> can be omitted. If you omit <NRf>, the most recent history parameter value will be queried. If you include <NRf>, the history parameter value of the waveform <NRf> times before the newest history waveform will be queried.
- After cyclic statistical processing has been completed. <NRf> can be omitted. If you omit <NRf>, the parameter value within the cycle that was measured last will be queried.
  If you include <NRf>, the parameter value within the cycle that was measured <NRf> times after the measurement shown on the screen’s left edge will be queried.
- When you execute a normal measurement, without cyclic statistical processing, even if you include <NRf> in the command, you cannot query the waveform parameters in the history memory.

:MEASure:{XY<x>}?

Function Queries all the settings for automated measurement of XY waveform parameters.

Syntax
```
:MEASure:{XY<x>}?
```

:MEASure:{XY<x>}:<Parameter>?

Function Queries all the settings of a waveform parameter (measurement item).

Syntax
```
:MEASure:{XY<x>}:<Parameter>?
XY<x>'s <x> = 1 to 8
<Parameter> = {XY1Integ|XY2Integ}
```

:MEASure:{XY<x>}:<Parameter>:COUNt?

Function Queries the count of measured values for cyclic statistical processing of a waveform parameter.

Syntax
```
:MEASure:{XY<x>}:<Parameter>:COUNt?
<Parameter> = {XY1Integ|XY2Integ}
```

Example
```
:MEASURE:XY1:XY1INTEG:COUNT? ->
:MEASURE:XY1:XY1INTEG:COUNT 1
```

:MEASure:{XY<x>}:<Parameter>:{MAXimum|MEAN|MINimum|SDEViation}?

Function Queries a cyclic statistical processing value of a waveform parameter.

Syntax
```
:MEASure:{XY<x>}:<Parameter>:
{MAXimum|MEAN|MINimum|SDEViation}?
<Parameter> = {XY1Integ|XY2Integ}
```

Example
```
:MEASURE:XY1:XY1INTEG:MAXIMUM? ->
:MEASURE:XY1:XY1INTEG:MAXIMUM NAN
```

Description This command returns “NAN” for any statistics that cannot be processed.

:MEASure:{XY<x>}:<Parameter>:STATe

Function Sets or queries whether the specified waveform’s waveform parameter (measurement item) is ON or OFF.

Syntax
```
:MEASure:{XY<x>}:<Parameter>:
STATe {<Boolean>}
:MEASure:{XY<x>}:<Parameter>:STATe?
<Parameter> = {XY1Integ|XY2Integ}
```

Example
```
:MEASURE:XY1:XY1INTEG:STATE ON
:MEASURE:XY1:XY1INTEG:STATE? ->
:MEASURE:XY1:XY1INTEG:STATE 1
```

:MEASure:{XY<x>}:<Parameter>:VALue?

Function Queries the measured value of a waveform parameter.

Syntax
```
:MEASure:{XY<x>}:<Parameter>:
VALue? {<NRf>}
XY<x>'s <x> = 1 to 8
<NRf> = 1 to 48000
<Parameter> = {XY1Integ|XY2Integ}
```

Description This command returns “NAN” if the value cannot be measured.

Parameters cannot be measured if the measurement has not been performed or if the measurement was performed, but computations were not performed because the specified range exceeded 10 Mpoint.

The <NRf> is used to specify which iteration of statistical processing to query the parameter value from. This command returns “NAN” (Not A Number) if the specified value does not exist.

- For non-cyclic statistical processing:
  <NRf> can be omitted. If you omit <NRf>, the most recent history parameter value will be queried.
  If you include <NRf>, the history parameter value of the waveform <NRf> times before the newest history waveform will be queried.
- After cyclic statistical processing has been completed:
  <NRf> can be omitted. If you omit <NRf>, the parameter value within the cycle that was measured last will be queried.
  If you include <NRf>, the parameter value within the cycle that was measured <NRf> times after the measurement shown on the screen’s left edge will be queried.

:MEASure:CYCLe?

Function Queries all cycle statistics settings.

Syntax
```
:MEASure:CYCLe?
```

:MEASure:CYCLe:ABORt

Function Aborts cyclic statistical processing.

Syntax
```
:MEASure:CYCLe:ABORt
```

Example
```
:MEASURE:CYCLE:ABORT
```

:MEASure:CYCLe:EXECute

Function Executes cyclic statistical processing.

Syntax
```
:MEASure:CYCLe:EXECute
```

Example
```
:MEASURE:CYCLE:EXECUTE
```

:MEASure:CYCLe:TRACe

Function Sets or queries the cycle trace of cycle statistics.

Syntax
```
:MEASure:CYCLe:TRACe {OWN|<NRf>
[,<NRf>]|MATH<x>}
:MEASure:CYCLe:TRACe?
First <NRf> = 1 to 16
Second <NRf> = 1 to 16 (sub channel)
MATH<x>'s <x> = 1 to 8
```

Example
```
:MEASURE:CYCLE:TRACE OWN
:MEASURE:CYCLE:TRACE? -> :MEASURE:
CYCLE:TRACE OWN
```

:MEASure:HISTory:ABORt

Function Aborts the statistical processing of history waveforms.

Syntax
```
:MEASure:HISTory:ABORt
```

Example
```
:MEASURE:HISTORY:ABORT
```

:MEASure:HISTory:EXECute

Function Executes the statistical processing of history waveforms.

Syntax
```
:MEASure:HISTory:EXECute
```

Example
```
:MEASURE:HISTORY:EXECUTE
```

:MEASure:MODE

Function Sets or queries the measure mode.

Syntax
```
:MEASure:MODE {OFF|ON|CYCLe|HISTory|
STATistics}
:MEASure:MODE?
```

Example
```
:MEASURE:MODE ON
:MEASURE:MODE? -> :MEASURE:MODE ON
```

:MEASure:ONECycle

Function Sets or queries whether one cycle mode is ON or OFF.

Syntax
```
:MEASure:ONECycle {<Boolean>}
:MEASure:ONECycle?
```

Example
```
:MEASURE:ONECYCLE ON
:MEASURE:ONECYCLE? -> :MEASURE:
ONECYCLE 1
```

:MEASure:TRANge (Time Range)

Function Sets or queries the waveform parameter measurement range.

Syntax
```
:MEASure:TRANge {<NRf>,<NRf>}
:MEASure:TRANge?
<NRf> = -5 to 5 (div; in 10 div/
display record length steps)
```

Example
```
:MEASURE:TRANGE -5,5
:MEASURE:TRANGE? -> :MEASURE:
TRANGE -5.000000000000,
5.000000000000
```

:MEASure:WAIT?

Function Waits for the completion of measurement by using a timeout valuer.

Syntax
```
:MEASure:WAIT? {<NRf>}
<NRf> = 1 to 36000 (timeout value,
in units of 100 msec)
```

Example
```
:MEASure:WAIT? 100 -> :MEAS:WAIT 1
```

Description The command returns 0 if the measurement finishes within the specified timeout. If measurement does not finish, or if it was never taking place to begin with, the command returns 1. Even if you make the timeout value long, 0 is returned as soon as the measurement finishes.

5.22 MONitor Group

The commands in this group deal with switching the screen that is displayed and the output of the numeric monitor.

:MONitor:ASENd?

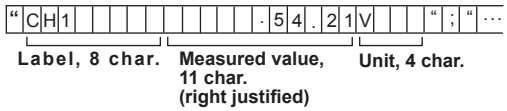
Function Sends the numeric monitor data (in ASCII format) of all channels.

Syntax `:MONitor:ASENd?`

Example `:MONITOR:ASEND? -> See the response format shown in the figure below.`

Description
- The measured values of each channel are divided by semicolons (0x3b). The values for all active channels are output.
- Depending on the MONitor:VERBose command's setting, the measured value is returned with a label and the unit, or just the measured value is returned.
- The active channels are those that meet the following conditions. For two-channel modules, they are the channels that are displayed. For 16-channel modules, they are the sub channels that are displayed and whose input is off.
- Logic channels follow the same guidelines explained here. The measured value is right justified. The unit is returned as white space.

Response Format



:MONitor:ASENd:CHANnel<x>[:SCHannel<x>]?

Function Sends the numeric monitor data (in ASCII format) of the specified channel.

Syntax
```
:MONitor:ASENd:
CHANnel<x>[:SCHannel<x>]?
CHANnel<x>'s <x> = 1 to 16
SCHannel<x>'s <x> = 1 to 16 (sub channel)
```

Example `:MONITOR:ASEND:CHAN1? -> See the response format for :MONitor:ASENd.`

Description
- Depending on the MONitor:VERBose command's setting, the measured value is returned with a label and the unit, or just the measured value is returned.
- If the specified channel is not displayed, "OFF" is returned for the measured value.

:MONitor:BITS:CHANnel<x>[:SCHannel<x>]?

Function Queries the effective bit length of the specified channel's data.

Syntax
```
:MONitor:BITS:CHANnel<x>[:
SCHannel<x>]?
CHANnel<x>'s <x> = 1 to 16
SCHannel<x>'s <x> = 1 to 16 (sub channel)
```

Example
```
:MONitor:BITS:CHAN1? -> :MONITOR
:BITS:CHAN1 16
```

Description For example, if this command is received by a 701250 voltage module (effective bit length of 12 bits), 12 is be returned. A logic module returns 8.

:MONitor:BYTeorder

Function Sets or queries the transmission byte order for data formats that are 2 bytes or longer.

Syntax
```
:MONitor:BYTeorder{LSBFirst|
MSBFirst}
:MONitor:BYTeorder?
```

Example
```
:MONITOR:BYTEORDER MSBFIRST
:MONITOR:BYTEORDER? -> :MONITOR
:BYTEORDER MSBFIRST
```

:MONitor:FORMat:CHANnel<x>[:SCHannel<x>]

Function Sets or queries the transmission data format.

Syntax
```
:MONitor:FORMat:
CHANnel<x>[:SCHannel<x>] {BYTE|WORD}
:MONitor:FORMat:
CHANnel<x>[:SCHannel<x>]?
CHANnel<x>'s <x> = 1 to 16
SCHannel<x>'s <x> = 1 to 16 (sub
channel)
```

Example
```
:MONITOR:FORMAT:CHANNEL1:
SCHANNEL1 WORD
:MONITOR:FORMAT:CHANNEL1:SCHANNEL1?
-> :MONITOR:FORMAT:CHANNEL1:
SCHANNEL1 WORD
```

:MONitor:OFFSet:CHANnel<x>[:SCHannel<x>]?

Function Queries the offset that is used when the specified channel's numeric monitor data is converted to a physical value.

Syntax
```
:MONitor:OFFSet:
CHANnel<x>[:SCHannel<x>]?
CHANnel<x>'s <x> = 1 to 16
SCHannel<x>'s <x> = 1 to 16 (sub
channel)
```

Example
```
:MONITOR:OFFSET:CHANNEL1 -> :MONITOR:
OFFSET:CHANNEL1 0.0000000E+00
```

:MONitor:GAIN:CHANnel<x>[:SCHannel<x>]?

Function Queries the gain that is used when the specified channel's numeric monitor data is converted to a physical value.

Syntax
```
:MONitor:GAIN:
CHANnel<x>[:SCHannel<x>]?
CHANnel<x>'s <x> = 1 to 16
SCHannel<x>'s <x> = 1 to 16 (sub
channel)
```

Example `:MONITOR:GAIN:CHANNEL1?`

:MONitor:RANGe:CHANnel<x>[:SCHannel<x>]?

Function Queries the range that is used when the specified channel's numeric monitor data is converted to a physical value.

Syntax
```
:MONitor:RANGe:
CHANnel<x>[:SCHannel<x>]?
CHANnel<x>'s <x> = 1 to 16
SCHannel<x>'s <x> = 1 to 16 (sub
channel)
```

Example
```
:MONITOR:RANGE:CHANNEL1? -> :MONITOR:
RANGE:CHANNEL1 5.0000000E+00
```

:MONitor:SEND:ALL?

Function Sends the numeric monitor data (in binary format).

Syntax
```
:MONitor:SEND:ALL?
```

Example
```
:MON:SEND:ALL? -> #9? (9-digit
number of bytes; data byte sequence)
```

Description
- The number of output bytes for each channel is based on the value specified by the :MONitor:FORMat:CHANnel<x> command.
- Channels whose input is off are not sent.

:MONitor:SEND:CHANnel<x>[:SCHannel<x>]?

Function Sends the numeric monitor data (in binary format) of the specified channel.

Syntax
```
:MONitor:SEND:
CHANnel<x>[:SCHannel<x>]?
CHANnel<x>'s <x> = 1 to 16
SCHannel<x>'s <x> = 1 to 16
```

Example
```
:MON:SEND:CHAN1? -> #9? (9-digit
number of bytes; data byte sequence)
```

Description
- The number of output bytes for each channel is based on the value specified by the :MONitor:FORMat:CHANnel<x> command.
- Channels whose input is off are not sent.

:MONitor:VERBose

Function Set whether to include labels and units in the response to the MONitor:ASENd? command.

Syntax
```
:MONitor:VERBose {<Boolean>}
:MONitor:VERBose?
```

Example
```
:MONITOR:VERBOSE ON
:MONITOR:VERBOSE? -> :MONITOR:
VERBOSE 1
```

:MONitor:LATCh:ASENd?

Function Sends the numeric monitor data (in ASCII format) of all channels.

Syntax
```
:MONitor:LATCh:ASENd?
```

Example
```
:MONITOR:LATCH:ASEND? -> The response
format is the same as the :MONitor:
ASENd? command.
```

Description The data that is present at the time when you execute the latch command is sent. If the latch command has not been executed, an undefined value is returned.

:MONitor:LATCh:ASENd:CHANnel<x>[:SCHannel<x>]?

Function Sends the numeric monitor data (in ASCII format) of the specified channel.

Syntax
```
:MONitor:LATCh:ASENd:
CHANnel<x>[:SCHannel<x>]?
CHANnel<x>'s <x> = 1 to 16
SCHannel<x>'s <x> = 1 to 16 (sub
channel)
```

Example
```
:MONITOR:LATCH:ASEND:CHANNEL1? ->
The response format is the same as
the :MONitor:ASENd:
CHANnel<x>[:SCHannel<x>]? command.
```

Description The data that is present at the time when you execute the latch command is sent. If the latch command has not been executed, an undefined value is returned.

:MONitor:LATCh:EXECute

Function Latches the monitor data.

Syntax
```
:MONitor:LATCh:EXECute
```

:MONitor:LATCh:SEND:{ALL|CHANnel<x>[:SCHannel<x>]}?

Function Sends numeric monitor data.

Syntax
```
:MONitor:LATCh:SEND:{ALL|
CHANnel<x>[:SCHannel<x>]}?
CHANnel<x>'s <x> = 1 to 16
SCHannel<x>'s <x> = 1 to 16 (sub
channel)
```

Example
```
:MON:LATCh:SEND:CHAN1? -> #9?
(9-digit number of bytes; data byte
sequence)
```

Description The data that is present at the time when you execute the latch command is sent.

5.23 MTRigger Group

The command in this group deals with manual triggers.

:MTRigger

Function Manually triggers the DL850/DL850V.

Syntax `:MTRigger`

Example `:MTRIGGER`

5.24 RECall Group

:RECall:SETup<x>:EXECute

Function Recalls setup data from an internal memory area.

Syntax
```
:RECall:SETup<x>:EXECute
<x> = 1 to 3
```

Example
```
:RECALL:SETUP1:EXECUTE
```

Description If you specify an area that does not contain setup data, an error occurs.

5.25 SEARch Group

The commands in this group deal with searching for and extracting patterns from acquired waveforms. You can perform the same operations and make the same settings and queries that you can by using the SHIFT+ZOOM keys on the front panel.

If the selectable range of the time axis is "<NRf> = –5 to 5div," the selectable range varies depending on settings such as the record length. For details, see "Selectable Range of Cursor Positions" under "Notes about Cursor Measurement" in chapter 9 of IM DL850-01EN.

:SEARch?

Function Queries all search settings.
Syntax `:SEARch?`

:SEARch:ABORt

Function Aborts the search.
Syntax `:SEARch:ABORt`
Example `:SEARCH:ABORT`

:SEARch:EDGE?

Function Queries all edge search settings.
Syntax `:SEARch:EDGE?`

:SEARch:EDGE:COUNt

Function Sets or queries the edge search count.
Syntax
```
:SEARch:EDGE:COUNt {<NRf>}
:SEARch:EDGE:COUNt?
<NRf> = 1 to 1000000
```
Example
```
:SEARCH:EDGE:COUNT 100
:SEARCH:EDGE:COUNT? -> :SEARCH:EDGE:
COUNT 100
```

:SEARch:EDGE:HYSTeresis

Function Sets or queries the edge search determination-level hysteresis.
Syntax
```
:SEARch:EDGE:HYSTeresis{HIGH|LOW|
MIDDle}
:SEARch:EDGE:HYSTeresis?
```
Example
```
:SEARCH:EDGE:HYSTERESIS LOW
:SEARCH:EDGE:HYSTERESIS? -> :SEARCH:
EDGE:HYSTERESIS LOW
```
Description This setting is valid if the channel that is being searched is not a logic channel.

:SEARch:EDGE:LEVEl

Function Sets or queries the edge search determination level.
Syntax
```
:SEARch:EDGE:LEVEl {<Voltage>|
<Current>|<NRf>}
:SEARch:EDGE:LEVEl?
```
Example
```
:SEARCH:EDGE:LEVEL 2
:SEARCH:EDGE:LEVEL? -> :SEARCH:EDGE:
LEVEL 2.00000E+00
```

:SEARch:EDGE:[LOGic:]BIT<x>

Function Sets or queries the edge polarity of the specified bit for when the edge search is performed on logic traces.
Syntax
```
:SEARch:EDGE:[LOGic:]
BIT<x> {BISLope|OFF|FALL|RISE}
:SEARch:EDGE:[LOGic:]BIT<x>?
<x> = 1 to 8
```
Example
```
:SEARCH:EDGE:LOGIC:BIT1 BISLOPE
:SEARCH:EDGE:LOGIC:BIT1? -> :SEARCH:
EDGE:LOGIC:BIT1 BISLOPE
```

:SEARch:EDGE:SLOPe

Function Sets or queries the edge polarity for when the edge search is performed on traces other than logic traces.
Syntax
```
:SEARch:EDGE:SLOPe {BISLope|FALL|
RISE}
:SEARch:EDGE:SLOPe?
```
Example
```
:SEARCH:EDGE:SLOPE RISE
:SEARCH:EDGE:SLOPE? -> :SEARCH:EDGE:
SLOPE RISE
```

:SEARch:EDGE:SOURce

Function Sets or queries the trace to perform the edge search on.
Syntax
```
:SEARch:EDGE:SOURce {<NRf>[,<NRf>]}
:SEARch:EDGE:SOURce?
<NRf> = 1 to 16
[,<NRf>] = If the channel has sub
channels, this is the range of sub
channel numbers.
```
Example
```
:SEARCH:EDGE:SOURCE 1
:SEARCH:EDGE:SOURCE? -> :SEARCH:
EDGE:SOURCE 1
```
Description If no module is installed, you cannot select the edge search.

:SEARch:EVENt?

Function Queries all event search settings.
Syntax `:SEARch:EVENt?`

:SEARch:EVENt:COUNt

Function Sets or queries the event search event number.

Syntax
```
:SEARch:EVENt:COUNt {<NRf>}
:SEARch:EVENt:COUNt?
<NRf> = 1 to 5000
```

Example
```
:SEARCH:EVENT:COUNT 1
:SEARCH:EVENT:COUNT? -> :SEARCH:
EVENT:COUNT 1
```

:SEARch:EVENt:SOURce

Function Sets or queries the event search source event.

Syntax
```
:SEARch:EVENt:SOURce {CAPTure|
MANual}
:SEARch:EVENt:SOURce?
```

Example
```
:SEARCH:EVENT:SOURCE CAPTURE
:SEARCH:EVENT:SOURCE? -> :SEARCH:
EVENT:SOURCE CAPTURE
```

:SEARch:EXECute

Function Executes the search. This is an overlap command.

Syntax
```
:SEARch:EXECute:
```

:SEARch:LSTate?

Function Queries all logic pattern search settings.

Syntax
```
:SEARch:LSTate?
```

:SEARch:LSTate:BIT<x>

Function Sets or queries the level of the specified bit for logic pattern searches.

Syntax
```
:SEARdh:LSTate:BIT<x> {DONTcare|
HIGH|LOW}
:SEARdh:LSTate:BIT<x>?
<x> = 1 to 8
```

Example
```
:SEARCH:LSTATE:BIT1 HIGH
:SEARCH:LSTATE:BIT1? -> :SEARCH:
LSTATE:BIT1 HIGH
```

:SEARch:LSTate:SOURce

Function Sets or queries the trace to perform the logic pattern search on.

Syntax
```
:SEARch:LSTate:
SOURce {<NRf> [,<NRf>]}
:SEARch:LSTate:SOURce?
<NRf> = 1 to 16
Second <NRf> = 1 to 16 (sub channel)
```

Example
```
:SEARCH:LSTATE:SOURCE 7
:SEARCH:LSTATE:SOURCE? -> :SEARCH:
LSTATE:SOURCE 7
```

:SEARch:LSTate:COUNt

Function Sets or queries the logic pattern search count.

Syntax
```
:SEARch:LSTate:COUNt {<NRf>}
:SEARch:LSTate:COUNt?
<NRf> = 1 to 1000000
```

Example
```
:SEARCH:LSTATE:COUNT 100
:SEARCH:LSTATE:COUNT? -> :SEARCH:
LSTATE:COUNT 100
```

Description This setting is the same as the edge search count setting.

:SEARch:SELect

Function Sets the search point that is displayed on the zoom window, and queries the zoom position of that search point.

Syntax
```
:SEARch:SELect {<NRf>|MAXimum}
:SEARch:SELect?
<NRf> = 1 to 5000
```

Example
```
:SEARCH:SELECT 4
:SEARCH:SELECT? -> :SEARCH:
SELECT -0.721900000000
```

Description The string “NAN” is returned if the search could not retrieve a position.

:SEARch:SELect? MAXimum

Function Queries the maximum save number, which is the number that is attached to the last position that the search retrieved.

Syntax
```
:SEARch:SELect? MAXimum
```

Example
```
:SEARCH:SELECT? MAXIMUM -> :SEARCH:
SELECT 5
```

Description The string “NAN” is returned if the search could not retrieve a position.

:SEARch:EPOint (End POint)

Function Sets or queries the search end position.

Syntax
```
:SEARch:EPOint {<NRf>}
:SEARch:EPOint?
<NRf> = −5 to 5div (the resolution
depends on the measurement length)
```

Example
```
:SEARCH:EPOINT 3
:SEARCH:EPOINT? -> :SEARCH:
EPOINT 3.000000000000
```

:SEARch:SPOint (Start POint)

Function Sets or queries the search start position.

Syntax
```
:SEARch:SPOint {<NRf>}
:SEARch:SPOint?
<NRf> = −5 to 5div (the resolution
depends on the measurement length)
```

Example
```
:SEARCH:SPOINT -5
:SEARCH:SPOINT? -> :SEARCH:
SPOINT -5.000000000000
```

:SEARch:TIME?

Function Queries all time search settings.

Syntax
```
:SEARch:TIME?
```

:SEARch:TIME:TIME

Function Sets or queries the time of the time search.

Syntax
```
:SEARch:TIME:TIME {<NRf>,<NRf>,
<NRf>,<NRf>,<NRf>,<NRf>,<NRf>}
:SEARch:TIME:TIME?
<NRf> = 2000 or greater (year)
<NRf> = 1 to 12 (month)
<NRf> = 1 to 31 (day)
<NRf> = 0 to 23 (hour)
<NRf> = 0 to 59 (minute)
<NRf> = 0 to 59 (second)
<NRf> = 0 to 999999 (microsecond)
```

Example
```
:SEARCH:TIME:TIME 2010,1,1,0,0,0,0
:SEARCH:TIME:TIME? -> :SEARCH:TIME
:TIME 2010,1,1,0,0,0,0
```

Description The initial value is the reference time of the waveform that is displayed when you open the menu. If there are no history waveforms or if the menu is not open, the initial value is 00:00:00:00 on January 1, 2010.

:SEARch:TWINdow (Target WINdow)

Function Sets or queries the window that search results will be displayed in.

Syntax
```
SEARch:TWINdow {Z1|Z2}
SEARch:TWINdow?
```

Example
```
:SEARCH:TWINDOW Z1
:SEARCH:TWINDOW? -> :SEARCH:
TWINDOW Z1
```

:SEARch:TYPE

Function Sets or queries the search type.

Syntax
```
:SEARch:TYPE {EDGE|EVENt|LSTate|
MMAX|TIME}
:SEARch:TYPE?
```

Example
```
:SEARCH:TYPE EDGE
:SEARCH:TYPE? -> :SEARCH:TYPE EDGE
```

:SEARch:MAG<x>

Function Sets or queries a zoom waveform magnification.

Syntax
```
:SEARch:MAG<x> {<NRf>}
:SEARch:MAG<x>?
<x> = 1, 2
<NRf> = 1 to 200000000
```

Example
```
:SEARCH:MAG1 1000
:SEARCH:MAG1? -> :SEARCH:MAG1 1000.0
```

Description This command is valid when the DL850/DL850V is using an external clock. When the DL850/DL850V is using the internal clock, use the :SEARch:TDIV<x> command to set the magnification.

:SEARch:POSition<x>

Function Sets or queries the position of a zoom box.

Syntax
```
:SEARch:POSition<x> {<NRf>}
:SEARch:POSition<x>?
<x> = 1, 2
<NRf> = -5 to +5 div
```

Example
```
:SEARCH:POSITION1 -0.7219
:SEARCH:POSITION1? -> :SEARCH:
POSITION1 -0.721900000000
```

:SEARch:TDIV<x>

Function Sets or queries a zoom waveform T/DIV value.

Syntax
```
:SEARch:TDIV<x>{<Time>|HOUR1|HOUR2|
HOUR3|HOUR4|HOUR5|HOUR6|HOUR7|HOUR8|
HOUR9|HOUR10|HOUR12|DAY1|DAY2|DAY3}
:SEARch:TDIV<x>?
<x> = 1, 2 (sets the target zoom
box)
<Time> = the same as the T/Div
setting
```

Example
```
:SEARCH:TDIV1 0.01
:SEARCH:TDIV1? -> :SEARCH:
TDIV1 10.00E-03
```

Description This command is valid when the DL850/DL850V is using the internal clock. When the DL850/DL850V is using an external clock, use the :SEARch:MAG<x> command to set the magnification.

5.26 SNAP Group

The command in this group is used to take snapshots. You can execute the same operation that you can by using the SNAP SHOT key on the front panel.

:SNAP

Function Takes a snapshot.

Syntax :SNAP

Example :SNAP

5.27 SSTart Group

The commands in this group deal with executing the single start operation. These commands set the trigger mode to Single and start waveform acquisition.

:SSTart

Function Executes the single start operation.
Syntax :SSTart
Example :SSTART

:SSTart? {<NRf>}

Function Executes the single start operation and waits for its completion by using a timeout valuer.
Syntax :SSTart? {<NRf>}
Example :SSTart? 100
Description <NRf> = 1 to 36000 (timeout value in 100 ms steps; start and wait)
–36000 to –1 (timeout value in 100 ms steps; wait without starting)

5.28 STARt Group

The command in this group is used to start waveform acquisition. You can execute the same operation that you can by using the START/STOP key on the front panel.

:STARt

Function Starts waveform acquisition.

Syntax :STRTt

Example :START

5.29 STATus Group

The commands in this group are used to make settings and queries related to the status report. There are no front panel keys that correspond to the commands in this group. For information about status reports, see chapter 6.

:STATus?

Function Queries all the settings for the communication status feature.

Syntax `:STATus?`

:STATus:CONDition?

Function Queries the contents of the condition register.

Syntax `:STATus:CONDition?`

Example `:STATUS:CONDITION? -> 16`

Description For details on how to use the :STATus:CONDition command to synchronize the DL850/DL850V, see page 4-9.

:STATus:EESE

Function Sets or queries the extended event enable register.

Syntax `:STATus:EESE <Register>`
`:STATus:EESE?`
`<Register> = 0 to 65535`

Example `:STATUS:EESE #B00000000`
`:STATUS:EESE? -> :STATUS:EESE 0`

:STATus:EESR?

Function Queries the contents of the extended event register and clears the register.

Syntax `:STATus:EESR?`

Example `:STATUS:EESR? -> 0`

:STATus:ERRor?

Function Queries the error code and message of the last error that has occurred.

Syntax `:STATus:ERRor?`

Description
- If no errors have occurred, 0,"No error" is returned.
- You can use the :STATus:QMESsage command to specify whether the message is included.

:STATus:FILTer<x>

Function Sets or queries the transition filter.

Syntax `:STATus:FILTer<x> {RISE|FALL|BOTH|NEVer}`
`:STATus:FILTer<x>?`
`<x> = 1 to 16`

Description Set how each bit in the condition register must change to trigger the setting of an event. If you specify RISE, a change from 0 to 1 triggers the setting of an event.

:STATus:QENable

Function Sets or queries whether messages other than errors will be stored to the error queue (ON) or not (OFF).

Syntax `:STATus:QENable {<Boolean>}`
`:STATus:QENable?`

Example `:STATUS:QENABLE ON`
`:STATUS:QENABLE? -> :STATUS:QENABLE 1`

:STATus:QMESsage

Function Sets or queries whether message information will be attached to the response to the STAT:ERR? query (ON) or not (OFF).

Syntax `:STATus:QMESsage {<Boolean>}`
`:STATus:QMESsage?`

Example `:STATUS:QMESSAGE ON`
`:STATUS:QMESSAGE? -> :STATUS:QMESSAGE 1`

5.30 STOP Group

The command in this group is used to stop waveform acquisition. You can execute the same operation that you can by using the START/STOP key on the front panel.

:STOP

Function Stops waveform acquisition.

Syntax :STOP

Description If recording is in progress, it is also stopped.

5.31 STORe Group

The commands in this group deal with saving setup data. You can perform the same operations that you can by pressing the SETUP key on the front panel, the Setup Data Store/Recall soft key, and then the Store Exec soft key.

:STORe?

Function Queries all the information related to setup data in the internal memory.

Syntax `:STORe?`

:STORe:SETup<x>?

Function Queries information about the setup data in the specified location of the internal memory.

Syntax `:STORe:SETup<x>?`
`<x> = 1 to 3`

:STORe:SETup<x>:COMMent

Function Sets or queries the comment for the setup data that is stored to the specified location in the internal memory.

Syntax `:STORe:SETup<x>:COMMent {<String>}`
`:STORe:SETup<x>:COMMent?`
`<x> = 1 to 3`
`<String> = Up to 14 characters`

Example `:STORE:SETUP1:COMMENT "ABCD"`
`:STORE:SETUP1:COMMENT? -> :STORE:SETUP1:COMMENT "ABCD"`

:STORe:SETup<x>:DATE?

Function Queries the date and time of the setup data that is stored to the specified location in the internal memory.

Syntax `:STORe:SETup<x>:DATE?`
`<x> = 1 to 3`

Example `:STORE:SETUP1:DATE? ->`
`:STORE:SETUP1:`
`DATE "2010/03/11 09:44:02"`

Description This command returns “No Valid Data” if no setup data is stored in the specified location.

:STORe:SETup<x>:EXECute

Function Saves setup data to the specified location in the internal memory.

Syntax `:STORe:SETup<x>:EXECute`
`<x> = 1 to 3`

Example `:STORE:SETUP1:EXECUTE`

5.32 SYSTem Group

The commands in this group deal with the system. You can make the same settings and queries that you can by pressing the UTILITY key on the front panel, and then using the System Config menu.

:SYSTem?

Function Queries all system settings.
Syntax :SYSTem?

:SYSTem:CLICk

Function Sets or queries whether click sounds are produced.
Syntax :SYSTem:CLICk {<Boolean>}
:SYSTem:CLICk?
Example :SYSTEM:CLICK ON
:SYSTEM:CLICK? -> :SYSTEM:CLICK 1

:SYSTem:CLOCk?

Function Queries all date/time settings.
Syntax :SYSTem:CLOCk?

:SYSTem:CLOCk:DATE

Function Sets or queries the date.
Syntax :SYSTem:CLOCk:DATE <String>
:SYSTem:CLOCk:DATE?
<String> = YYYY/MM/DD Specify the year according to the Gregorian calendar.
YYYY: 2000 to 2099
Example :SYSTEM:CLOCK:DATE "2010/04/01"
:SYSTEM:CLOCK:DATE? -> :SYSTEM:CLOCK:DATE "2010/04/01"
Description Regardless of the setting that you have made using the SYST:CLOC:FORM command, this command returns responses in the format shown above.

:SYSTem:CLOCk:FORMat

Function Sets or queries the date format.
Syntax :SYSTem:CLOCk:FORMat {<NRf>}
:SYSTem:CLOCk:FORMat?
<NRf> = 1 to 4
Example :SYSTEM:CLOCK:FORMAT 1
:SYSTEM:CLOCK:FORMAT? -> :SYSTEM:CLOCK:FORMAT 1
Description This command specifies the display format of the date. (The communication response format of the date is fixed.)
1: Year/month (number)/day
2: Day/month (number)/year
3: Day-month (English abbreviation)-year (last two digits)
4: Day month (English abbreviation) year

:SYSTem:CLOCk:MODE

Function Sets or queries whether the date and time are displayed.
Syntax :SYSTem:CLOCk:MODE {<Boolean>}
:SYSTem:CLOCk:MODE?
Example :SYSTEM:CLOCK:MODE ON
:SYSTEM:CLOCK:MODE? -> :SYSTEM:CLOCK:MODE 1

:SYSTem:CLOCk:SNTP?

Function Queries all SNTP settings.
Syntax :SYSTem:CLOCk:SNTP?

:SYSTem:CLOCk:SNTP:EXECute

Function Uses SNTP to set the date and time.
Syntax :SYSTem:CLOCk:SNTP:EXECute
Example :SYSTEM:CLOCK:SNTP:EXECUTE

:SYSTem:CLOCk:SNTP:GMTTime

Function Sets or queries the time difference from GMT when SNTP is being used.
Syntax :SYSTem:CLOCk:SNTP:GMTTime {<String>}
:SYSTem:CLOCk: SNTP:GMTTime?
<String> = HH:MM (-12:00 to 12:00)
Example :SYSTEM:CLOCK:SNTP:GMTTIME "09:00"
:SYSTEM:CLOCK:SNTP:GMTTIME? -> :SYSTEM:CLOCK:SNTP:GMTTIME "09:00"

:SYSTem:CLOCk:TIME

Function Sets or queries the time.
Syntax :SYSTem:CLOCk:TIME <String>
:SYSTem:CLOCk:TIME?
<String> = HH:MM:SS HH can be set to a value from 0 to 23.
Example :SYSTEM:CLOCK:TIME "10:00:00"
:SYSTEM:CLOCK:TIME? -> :SYSTEM:CLOCK:TIME "10:00:00"

:SYSTem:CLOCk:TYPE

Function Sets or queries whether the date and time are set manually or by using SNTP.
Syntax :SYSTem:CLOCk:TYPE {MANual|SNTP}
:SYSTem:CLOCk:TYPE?
Example :SYSTEM:CLOCK:TYPE MANUAL
:SYSTEM:CLOCK:TYPE? -> :SYSTEM:CLOCK:TYPE MANUAL

:SYSTem:IRIG?

Function Queries all IRIG settings.

Syntax
```
:SYSTem:IRIG?
```

:SYSTem:IRIG:FORMat

Function Sets or queries the IRIG format for synchronizing the time through IRIG signals.

Syntax
```
:SYSTem:IRIG:FORMat {A|B}
:SYSTem:IRIG:FORMat?
```

Example
```
:SYSTEM:IRIG:FORMAT A
:SYSTEM:IRIG:FORMAT? -> :SYSTEM:
IRIG:FORMAT A
```

Description This command is valid on models with the /C20 option.

:SYSTem:IRIG:MODulation

Function Sets or queries the IRIG modulation type for synchronizing the time through IRIG signals.

Syntax
```
:SYSTem:IRIG:MODulation {AM|PWCode}
:SYSTem:IRIG:MODulation?
```

Example
```
:SYSTEM:IRIG:MODULATION AM
:SYSTEM:IRIG:MODULATION? -> :SYSTEM:
IRIG:MODULATION AM
```

Description This command is valid on models with the /C20 option.

:SYSTem:IRIG:IMPedance

Function Sets or queries the input impedance for synchronizing the time through IRIG signals.

Syntax
```
:SYSTem:IRIG:IMPedance {<NRf>}
:SYSTem:IRIG:IMPedance?
<NRf> = 50 or 5000
```

Example
```
:SYSTEM:IRIG:IMPEDANCE 50
:SYSTEM:IRIG:IMPEDANCE? -> :SYSTEM:
IRIG:IMPEDANCE 50
```

Description This command is valid on models with the /C20 option.

:SYSTem:KEYResponse

Function Sets or queries the response time of the START/STOP key.

Syntax
```
:SYSTem:KEYResponse {QUICK|SEC1}
:SYSTem:KEYResponse?
```

Example
```
:SYSTEM:KEYRESPONSE SEC1
:SYSTEM:KEYRESPONSE? -> :SYSTEM:
KEYRESPONSE SEC1
```

:SYSTem:KEYProtect?

Function Queries all key lock settings.

Syntax
```
:SYSTem:KEYProtect?
```

:SYSTem:KEYProtect:MODE

Function Sets or queries whether the keys are locked.

Syntax
```
:SYSTem:KEYProtect:MODE {<Boolean>}
:SYSTem:KEYProtect:MODE?
```

Example
```
:SYSTEM:KEYPROTECT:MODE ON
:SYSTEM:KEYPROTECT:MODE? -> :SYSTEM:
KEYPROTECT:MODE 1
```

Description The key lock status is retained even when the DL850/DL850V is restarted.

:SYSTem:KEYProtect:TYPE

Function Sets or queries which keys will be locked.

Syntax
```
:SYSTem:KEYProtect:TYPE {ALL|ESTart}
:SYSTem:KEYProtect:TYPE?
```

Example
```
:SYSTEM:KEYPROTECT:TYPE ESTART
:SYSTEM:KEYPROTECT:TYPE? -> :SYSTEM:
KEYPROTECT:TYPE ESTART
```

:SYSTem:KEYProtect:RTYPe (Release TYPe)

Function Sets or queries how to release the key lock.

Syntax
```
:SYSTem:KEYProtect:RTYPe {KEY|
PASSword}
:SYSTem:KEYProtect:RTYPe?
```

Example
```
:SYSTEM:KEYPROTECT:RTYPE PASSWORD
:SYSTEM:KEYPROTECT:RTYPE? ->
:SYSTEM:KEYPROTECT:RTYPE PASSWORD
```

:SYSTem:KEYProtect:EPASsword (Enter password)

Function Enters the password to release the key lock.

Syntax
```
:SYSTem:KEYProtect:EPASsword
{<String>}
<String> = Up to 8 characters
```

Example
```
:SYSTEM:KEYPROTECT:EPASSWORD" ABC"
```

Description This command is valid when the keys are locked, a password is required to release the key lock, and a password has been set.

:SYSTem:KEYProtect:SPASsword (Set password)

Function Sets the password that is used to release the key lock.

Syntax
```
:SYSTem:KEYProtect:SPASsword
{<String>}
<String> = Up to 8 characters
```

Example
```
:SYSTEM:KEYPROTECT:SPASSWORD" ABC"
```

Description Send the command :SYSTem:KEYProtect: SPASsword "" to reset the password.

:SYSTem:LANGuage
Function Sets or queries the message language.
Syntax
```
:SYSTem:LANGuage {CHINese|ENGLish|
FRENch|GERMan|ITALian|JAPANese|
KOREan|SPANish}
:SYSTem:LANGuage?
```
Example
```
:SYSTEM:LANGUAGE ENGLISH
:SYSTEM:LANGUAGE? -> :SYSTEM:
LANGUAGE ENGLISH
```

:SYSTem:LCD?
Function Queries all LCD settings.
Syntax
```
:SYSTem:LCD?
```

:SYSTem:LCD:AUTO?
Function Queries all the settings for the feature that automatically turns off the backlight.
Syntax
```
:SYSTem:LCD:AUTO?
```

:SYSTem:LCD:AUTO:MODE
Function Sets or queries whether the feature that automatically turns off the backlight is on.
Syntax
```
:SYSTem:LCD:AUTO:MODE {<Boolean>}
:SYSTem:LCD:AUTO:MODE?
```
Example
```
:SYSTEM:LCD:AUTO:MODE ON
:SYSTEM:LCD:AUTO:MODE? -> :SYSTEM:
LCD:AUTO:MODE 1
```

:SYSTem:LCD:AUTO:TIME
Function Sets or queries the amount of time until the backlight is turned off.
Syntax
```
:SYSTem:LCD:AUTO:TIME {<NRf>}
:SYSTem:LCD:AUTO:TIME?
<NRf> = 1 to 60 (minute)
```
Example
```
:SYSTEM:LCD:AUTO:TIME 10
:SYSTEM:LCD:AUTO:TIME? -> :SYSTEM:
LCD:AUTO:TIME 10
```

:SYSTem:LCD:BRIGhtness
Function Sets or queries the LCD brightness.
Syntax
```
:SYSTem:LCD:BRIGhtness {<NRf>}
:SYSTem:LCD:BRIGhtness?
<NRf> = 1 to 10
```
Example
```
:SYSTEM:LCD:BRIGHTNESS 3
:SYSTEM:LCD:BRIGHTNESS? -> :SYSTEM:
LCD:BRIGHTNESS 3
```

:SYSTem:LCD:MODE
Function Sets or queries whether the backlight is on.
Syntax
```
:SYSTem:LCD:MODE {<Boolean>}
:SYSTem:LCD:MODE?
```
Example
```
:SYSTEM:LCD:MODE ON
:SYSTEM:LCD:MODE? -> :SYSTEM:LCD:
MODE 1
```

:SYSTem:LOGic?
Function Queries all logic display settings.
Syntax
```
:SYSTem:LOGic?
```

:SYSTem:LOGic:NFORmat (Numerical Format)
Function Sets or queries the numeric display format of logic module data.
Syntax
```
:SYSTem:LOGic:NFORmat {BIT|
HEXadecimal}
:SYSTem:LOGic:NFORmat?
```
Example
```
:SYSTEM:LOGIC:NFORMAT BIT
:SYSTEM:LOGIC:NFORMAT? -> :SYSTEM:
LOGIC:NFORMAT BIT
```
Description This setting is applied to all channels that have modules installed in them.

:SYSTem:LOGic:CORDer (Cursor Order)
Function Sets or queries the bit order that is used when you use cursors to display logic module data as bits.
Syntax
```
:SYSTem:LOGic:CORDer {LSBFirst|
MSBFirst}
:SYSTem:LOGic:CORDer?
```
Example
```
:SYSTEM:LOGIC:CORDER LSBFIRST
:SYSTEM:LOGIC:CORDER? -> :SYSTEM:
LOGIC:CORDER LSBFIRST
```
Description This setting is applied to all channels that have modules installed in them.

:SYSTem:LOGic:BORDer (Bit Order)
Function Sets or queries the bit order that is used when you display logic module data as waveforms.
Syntax
```
:SYSTem:LOGic:BORDer {LSBFirst|
MSBFirst}
:SYSTem:LOGic:BORDer?
```
Example
```
:SYSTEM:LOGIC:BORDER LSBFIRST
:SYSTEM:LOGIC:BORDER? -> :SYSTEM:
LOGIC:BORDER LSBFIRST
```
Description This setting is applied to all channels that have modules installed in them.

:SYSTem:MLANguage

Function Sets or queries the menu language.

Syntax
```
:SYSTem:MLANguage {CHINese|ENGLish|
FRENch|GERMan|ITALian|JAPANese|
KORean|SPANish}
:SYSTem:MLANguage?
```

Example
```
:SYSTEM:MLANGUAGE ENGLISH
:SYSTEM:MLANGUAGE? -> :SYSTEM:
MLANGUAGE ENGLISH
```

:SYSTem:OVERview

Function Displays the system information.

Syntax
```
:SYSTem:OVERview
```

Example
```
:SYSTEM:OVERVIEW
```

:SYSTem:PACTion

Function Sets or queries whether the action-on-trigger mode will be enabled when the DL850/DL850V turns on (ON) or not (OFF).

Syntax
```
:SYSTem:PACTion {<Boolean>}
:SYSTem:PACTion?
```

Example
```
:SYSTEM:PACTION ON
:SYSTEM:PACTION? -> :SYSTEM:
PACTION 1
```

:SYSTem:PSTart

Function Sets or queries whether waveform acquisition will be started when the DL850/DL850V turns on (ON) or not (OFF).

Syntax
```
:SYSTem:PSTart {<Boolean>}
:SYSTem:PSTart?
```

Example
```
:SYSTEM:PSTART ON
:SYSTEM:PSTART? -> :SYSTEM:PSTART 1
```

:SYSTem:RCMode (Remote Control Mode)

Function Sets or queries whether remote signals to stop measurements will be ignored (ON) or not (OFF).

Syntax
```
:SYSTem:RCMode {<Boolean>}
:SYSTem:RCMode?
```

Example
```
:SYSTEM:RCMODE ON
:SYSTEM:RCMODE? -> :SYSTEM:RCMODE 1
```

:SYSTem:SCALefont

Function Sets or queries the font size that is used for waveform labels and scale values.

Syntax
```
:SYSTem:SCALefont {SMALl|LARGe}
:SYSTem:SCALefont?
```

Example
```
:SYSTEM:SCALEFONT SMALL
:SYSTEM:SCALEFONT? -> :SYSTEM:
SCALEFONT SMALL
```

Description Even if you specify LARGe with this command, depending on the display conditions, the SMALl font may be displayed.

:SYSTem:SOITem (Scale On ITEM)

Function Sets or queries which scales are displayed.

Syntax
```
:SYSTem:SOITem {ALL|TSCale}
:SYSTem:SOITem?
```

Example
```
:SYSTEM:SOITEM ALL
:SYSTEM:SOITEM? -> :SYSTEM:
SOITEM ALL
```

:SYSTem:STORage:MEDia

Function Sets or queries the medium that you want to format.

Syntax
```
:SYSTem:STORage:MEDia {HD|SD|UUSB|
LUSB}
:SYSTem:STORage:MEDia?
```

Example
```
:SYSTEM:STORAGE:MEDIA SD
:SYSTEM:STORAGE:MEDIA? -> :SYSTEM:
STORAGE:MEDIA SD
```

Description HD: Internal hard disk
SD: SD memory card
UUSB|LUSB: USB storage medium

:SYSTem:STORage:FORMat:EXECute

Function Formats the specified medium. This is an overlap command.

Syntax
```
:SYSTem:STORage:FORMat:EXECute
```

Example
```
:SYSTEM:STORAGE:FORMAT:EXECUTE
```

:SYSTem:TSYNchro (Time Synchronization)

Function Sets or queries the time synchronization.

Syntax
```
:SYSTem:TSYNchro {IRIG|OFF}
:SYSTem:TSYNchro?
```

Example
```
:SYSTEM*TSYNCHRO IRIG
:SYSTEM*TSYNCHRO? ->
:SYSTEM*TSYNCHRO IRIG
```

Description This command is valid on models with the /C20 option.

:SYSTem:USBKeyboard

Function Sets or queries the USB keyboard type.

Syntax
```
:SYSTem:USBKeyboard {ENGLish|
JAPANese}
:SYSTem:USBKeyboard?
```

Example
```
:SYSTEM:USBKEYBOARD ENGLISH
:SYSTEM:USBKEYBOARD? -> :SYSTEM:
USBKEYBOARD ENGLISH
```

5.33 TIMebase Group

The commands in this group deal with time bases. You can make the same settings and queries that you can by using the TIME/DIV knob on the front panel.

:TIMebase?

Function Queries all time base settings.

Syntax
```
:TIMebase?
```

:TIMebase:CHANnel<x>:SRATe?

Function Queries the sample rate of the specified channel.

Syntax
```
:TIMebase:CHANnel<x>:SRATe?
:TIMebase:CHANnel<x>:SRATe?
<x> = 1 to 16
```

Example
```
:TIMEBASE:CHANNEL1:SRATE? ->
:TIMEBASE:CHANNEL1:
SRATE 1.0000000E+06
```

Description This command returns the current setting (the reserved value for the next measurement). Use the :WAVeform:CAPTure:SRATe? command to query the measured data.

:TIMebase:SOURce

Function Sets or queries the time base.

Syntax
```
:TIMebase:SOURce {EXTernal|INTernal}
:TIMebase:SOURce?
```

Example
```
:TIMEBASE:SOURCE INTERNAL
:TIMEBASE:SOURCE? -> :TIMEBASE:
SOURCE INTERNAL
```

:TIMebase:SRATe

Function Sets or queries the sample rate.

Syntax
```
:TIMebase:SRATE {<Frequency>}
:TIMebase:SRATE?
<Frequency> = 5Hz, 10Hz, 20Hz, 50Hz,
100Hz, ... 20MHz, 50MHz, 100MHz
```

Example
```
:TIMEBASE:SRATE 500kHz
:TIMEBASE:SRATE? -> :TIMEBASE:
SRATE 500.00000E+03
```

:TIMebase:TDIV

Function Sets or queries the T/DIV value.

Syntax
```
:TIMebase:TDIV {<Time>|HOUR1|HOUR2|
HOUR3|HOUR4|HOUR5|HOUR6|HOUR7|HOUR8|
HOUR9|HOUR10|HOUR12|DAY1|DAY2|DAY3}
:TIMebase:TDIV?
<Time> = 100ns to 1800s
```

Example
```
:TIMEBASE:TDIV 1ms
:TIMEBASE:TDIV? -> :TIMEBASE:
TDIV 1.000E-03
```

5.34 TRIGger Group

The commands in this group deal with triggers. You can make the same settings and queries that you can by using the TRIGGER group keys (MODE, SIMPLE, ENHANCED, POSITION, and DELAY keys) on the front panel.

:TRIGger?

Function Queries all trigger settings.

Syntax
```
:TRIGger?
```

:TRIGger:ABN? (A -> B(n))

Function Queries all A->B(n) trigger settings.

Syntax
```
:TRIGger:ABN?
```

:TRIGger:ABN:COUNt

Function Sets or queries the number of times condition B must be met for A->B(n) triggers.

Syntax
```
:TRIGger:ABN:COUNt {<NRf>}
:TRIGger:ABN:COUNt?
<NRf> = 1 to 10000
```

Example
```
:TRIGGER:ABN:COUNT 100
:TRIGGER:ABN:COUNT? -> :TRIGGER:ABN:
COUNT 100
```

:TRIGger:ACTion?

Function Queries all action-on-trigger settings.

Syntax
```
:TRIGger:ACTion?
```

:TRIGger:ACTion:BUZZer

Function Sets or queries whether a beep is sounded as an action.

Syntax
```
:TRIGger:ACTion:BUZZer {<Boolean>}
:TRIGger:ACTion:BUZZer?
```

Example
```
:TRIGGER:ACTION:BUZZER ON
:TRIGGER:ACTION:BUZZER? -> :TRIGGER:
ACTION:BUZZER 1
```

:TRIGger:ACTion:HCOPy

Function Sets or queries whether an screen capture is printed from the built-in printer as an action.

Syntax
```
:TRIGger:ACTion:HCOPy {<Boolean>}
:TRIGger:ACTion:HCOPy?
```

Example
```
:TRIGGER:ACTION:HCOPY ON
:TRIGGER:ACTION:HCOPY? -> :TRIGGER:
ACTION:HCOPY 1
```

:TRIGger:ACTion:IMAGe?

Function Queries all settings for saving screen captures as an action-on-trigger.

Syntax
```
:TRIGger:ACTion:IMAGe?
```

:TRIGger:ACTion:IMAGe[:MODE]

Function Sets or queries whether a screen capture is saved to the storage medium as an action.

Syntax
```
:TRIGger:ACTion:
IMAGe[:MODE] {<Boolean>}
:TRIGger:ACTion:IMAGe?
```

Example
```
:TRIGGER:ACTION:IMAGE:MODE ON
:TRIGGER:ACTION:IMAGE:MODE? ->
:TRIGGER:ACTION:IMAGE:MODE 1
```

:TRIGger:ACTion:IMAGe:CDIRectory

Function Sets or queries the current directory on the storage medium where screen captures are saved to as an action.

Syntax
```
:TRIGger:ACTion:IMAGe:
CDIRectory {<String>}
:TRIGger:ACTion:IMAGe:CDIRectory?
```

Example
```
:TRIGGER*ACTION:IMAGE:
CDIRECTORY "NO_1"
:TRIGGER*ACTION:IMAGE:CDIRECTORY? ->
:TRIGGER*ACTION:IMAGE:
CDIRECTORY "NO_1"
```

:TRIGger:ACTion:IMAGe:DRIVe

Function Sets the storage medium that screen captures are saved to as an action.

Syntax
```
:TRIGger:ACTion:IMAGe:
DRIVe {HD,<NRf>|NETWork|SD|
USB,<NRf>}
```

Example
```
:TRIGGER:ACTION:IMAG:DRIVE SD
```

:TRIGger:ACTion:IMAGe:PATH?

Function Queries the path on the storage medium where screen captures are saved to as an action.

Syntax
```
:TRIGger:ACTion:IMAGe:PATH?
```

Example
```
:TRIGger:ACTion:IMAGe:PATH? ->
:TRIG:ACT:IMAG:
PATH" Path=HD,1/20090506"
```

:TRIGger:ACTion:MAIL?

Function Queries all settings for sending e-mail as an action-on-trigger.

Syntax
```
:TRIGger:ACTion:MAIL?
```

:TRIGger:ACTion:MAIL:COUNt

Function Sets or queries the e-mail transmission limit for when e-mail is sent as an action.

Syntax
```
:TRIGger:ACTion:MAIL:COUNt {<NRf>}
:TRIGger:ACTion:MAIL:COUNt?
<NRf> = 1 to 1000
```

Example
```
:TRIGGER:ACTION:MAIL:COUNT 100
:TRIGGER:ACTION:MAIL:COUNT? ->
:TRIGGER:ACTION:MAIL:COUNT 100
```

:TRIGger:ACTion:MAIL:MODE

Function Sets or queries whether e-mail is sent as an action.

Syntax
```
:TRIGger:ACTion:MAIL:
MODE {<Boolean>}
:TRIGger:ACTion:MAIL:MODE?
```

Example
```
:TRIGGER:ACTION:MAIL:MODE ON
:TRIGGER:ACTION:MAIL:MODE? ->
:TRIGGER:ACTION:MAIL:MODE 1
```

:TRIGger:ACTion:MODE

Function Sets or queries whether action-on-trigger is used.

Syntax
```
:TRIGger:ACTion:MODE {<Boolean>}
:TRIGger:ACTion:MODE?
```

Example
```
:TRIGGER:ACTION:MODE ON
:TRIGGER:ACTION:MODE? -> :TRIGGER:
ACTION:MODE 1
```

:TRIGger:ACTion:SAVE?

Function Queries all the settings related to saving data as an action-on-trigger.

Syntax
```
:TRIGger:ACTion:SAVE?
```

:TRIGger:ACTion:SAVE[:MODE]

Function Sets or queries whether waveform data is saved to the storage medium as an action.

Syntax
```
:TRIGger:ACTion:
SAVE[:MODE] {<Boolean>}
:TRIGger:ACTion:SAVE:MODE?
```

Example
```
:TRIGGER:ACTION:SAVE:MODE ON
:TRIGGER:ACTION:SAVE:MODE? ->
:TRIGGER:ACTION:SAVE:MODE 1
```

:TRIGger:ACTion:SAVE:TYPE

Function Sets or queries the type of data to save when waveform data is saved to the storage medium as an action.

Syntax
```
:TRIGger:ACTion:SAVE:TYPE {ASCii|
BINary|FLOat}
:TRIGger:ACTion:SAVE:TYPE?
```

Example
```
:TRIGGER:ACTION:SAVE:TYPE BINARY
:TRIGGER:ACTION:SAVE:TYPE? ->
:TRIGGER:ACTION:SAVE:TYPE BINARY
```

:TRIGger:ACTion:SAVE:CDIRectory

Function Changes the current directory on the storage medium that waveform data is saved to as an action.

Syntax
```
:TRIGger:ACTion:SAVE:CDIRectory
{<String>}
```

Example
```
:TRIGGER:ACTION:SAVE:
CDIRECTORY "NO_1"
```

:TRIGger:ACTion:SAVE:DRIVe

Function Sets the storage medium that waveform data is saved to as an action.

Syntax
```
:TRIGger:ACTion:SAVE:DRIVe {HD,<NRf>|
NETWork|SD|USB,<NRf>}
```

Example
```
:TRIGGER:ACTION:SAVE:DRIVE SD
```

:TRIGger:ACTion:SAVE:PATH?

Function Queries the path on the storage medium that waveform data is saved to as an action.

Syntax
```
:TRIGger:ACTion:SAVE:PATH?
```

Example
```
:TRIGger:ACTion:SAVE:PATH? -> :TRIG:
ACT:SAVE:PATH" Path=HD,1/20090506"
```

:TRIGger:ADB? (A Delay B)

Function Queries all A Delay B trigger settings.

Syntax
```
:TRIGger:ADB?
```

:TRIGger:ADB:DELay

Function Sets or queries the delay time for condition B for A Delay B triggers.

Syntax
```
:TRIGger:ADB:DELay {<Time>}
:TRIGger:ADB:DELay?
<Time> = 0ns to 10s
```

Example
```
:TRIGGER:ADB:DEL 0.001
:TRIGGER:ADB:DEL? -> :TRIGGER:ADB:
DELAY 1.0000000E-03
```

Description The resolution is 10 ns.

:TRIGger:AND?

Function Queries all AND trigger settings.

Syntax
```
:TRIGger:AND?
```

:TRIGger:AND:CHANnel<x>:BIT<x>

Function Sets or queries the state of the specified bit of the specified logic channel for AND triggers.

Syntax
```
:TRIGger:AND:CHANnel<x>:
BIT<x> {DONTcare|HIGH|LOW}
:TRIGger:AND:CHANnel<x>:BIT<x>?
```

Example
```
:TRIGGER:AND:CHANNEL9:BIT1 DONTCARE
:TRIGGER:AND:CHANNEL9:BIT1? ->
:TRIGGER:AND:CHANNEL9:BIT1 DONTCARE
```

:TRIGger:AND:CHANnel<x>:SCHannel<x>:BIT<x>

Function Sets or queries the state of the specified bit of the specified sub channel of the specified logic channel for AND triggers.

Syntax
```
:TRIGger:AND:CHANnel<x>:SCHannel<x>:
BIT<x> {DONTcare|HIGH|LOW}
:TRIGger:AND:CHANnel<x>:SCHannel<x>:
BIT<x>?
```

Example
```
:TRIGGER:AND:CHANNEL1:SCHANNEL1:
BIT1 LOW
:TRIGGER:AND:CHANNEL1:SCHANNEL1:
BIT1? -> :TRIGGER:AND:CHANNEL1:
SCHANNEL1:BIT1 LOW
```

:TRIGger:AND:CHANnel<x>[:CONDition]

Function Sets or queries the state of the specified channel of the specified logic channel for AND triggers.

Syntax
```
:TRIGger:AND:CHANnel<x>[:CONDition]
{DONTcare|HIGH|LOW|WLIn|WLOut}
:TRIGger:AND:CHANnel<x>[:CONDition]?
```

Example
```
:TRIGGER:AND:CHANNEL2:CONDITION HIGH
:TRIGGER:AND:CHANNEL2:CONDITION? ->
:TRIGGER:AND:CHANNEL2:CONDITION HIGH
```

Description Use this command on channels that do not have sub channels.

:TRIGger:AND:CHANnel<x>:SCHannel<x>[:CONDition]

Function Sets or queries the state of the specified sub channel of the specified logic channel for AND triggers.

Syntax
```
:TRIGger:AND:CHANnel<x>:
SCHannel<x>[:CONDition] {DONTcare|
HIGH|LOW|WLIn|WLOut}
:TRIGger:AND:CHANnel<x>:
SCHannel<x>[:CONDition]?
```

Example
```
:TRIGGER:AND:CHANNEL15:SCHANNEL1:
CONDITION HIGH
:TRIGGER:AND:CHANNEL15:SCHANNEL1:
CONDITION? -> :TRIGGER:AND:CHANNEL15:
SCHANNEL1:CONDITION HIGH
```

Description Use this command on channels that have sub channels.

:TRIGger:ATRigger?

Function Queries all condition A settings.

Syntax
```
:TRIGger:ATRigger?
```

:TRIGger:ATRigger:CHANnel<x>:BIT<x>

Function Sets or queries the state of the specified bit of the specified logic channel of condition A.

Syntax
```
:TRIGger:ATRigger:CHANnel<x>:
BIT<x> {DONTcare|HIGH|LOW}
:TRIGger:ATRigger:CHANnel<x>:BIT<x>?
```

Example
```
:TRIGGER:ATRIGGER:CHANNEL9:BIT1 HIGH
:TRIGGER:ATRIGGER:CHANNEL9:BIT1? ->
:TRIGGER:ATRIGGER:CHANNEL9:BIT1 HIGH
```

:TRIGger:ATRigger:CHANnel<x>:SCHannel<x>:BIT<x>

Function Sets or queries the state of the specified bit of the specified sub channel of the specified logic channel of condition A.

Syntax
```
:TRIGger:ATRigger:CHANnel<x>:
SCHannel<x>:BIT<x> {DONTcare|HIGH|
LOW}
:TRIGger:ATRigger:CHANnel<x>:
SCHannel<x>:BIT<x>?
```

Example
```
:TRIGGER:ATRIGGER:CHANNEL1:
SCHANNEL1:BIT1 LOW
:TRIGGER:ATRIGGER:CHANNEL1:
SCHANNEL1:BIT1? -> :TRIGGER:ATRIGGER:
CHANNEL1:SCHANNEL1:BIT1 LOW
```

:TRIGger:ATRigger:CHANnel<x>[:CONDition]

Function Sets or queries the state of the specified channel of the specified logic channel of condition A.

Syntax
```
:TRIGger:ATRigger:
CHANnel<x>[:CONDition] {DONTcare|
HIGH|LOW}
:TRIGger:ATRigger:
CHANnel<x>[:CONDition]?
CHAN<x>'s <x> = 1 to 16
```

Example
```
:TRIGGER:ATRIGGER:CHANNEL1:
CONDITION HIGH
:TRIGGER:ATRIGGER:CHANNEL1:
CONDITION? -> :TRIGGER:ATRIGGER:
CHANNEL1:CONDITION HIGH
```

Description Use this command on channels that do not have sub channels.

:TRIGger:ATRigger:CHANnel<x>:SCHannel<x>[:CONDition]

Function Sets or queries the state of the specified sub channel of the specified logic channel of condition A.

Syntax
```
:TRIGger:ATRigger:CHANnel<x>:
SCHannel<x>[:CONDition] {DONTcare|
HIGH|LOW}
:TRIGger:ATRigger:CHANnel<x>:
SCHannel<x>[:CONDition]?
CHAN<x>'s <x> = 1 to 16
SCHannel<x>'s <x> = 1 to 16 (sub
channel)
```

Example
```
:TRIGGER:ATRIGGER:CHANNEL15:
SCHANNEL1:CONDITION HIGH
:TRIGGER:ATRIGGER:CHANNEL15:
SCHANNEL1:CONDITION? -> :TRIGGER
:ATRIGGER:CHANNEL15:SCHANNEL1:
CONDITION HIGH
```

Description Use this command on channels that have sub channels.

:TRIGger:ATRigger:CONDition

Function Sets or queries the achievement condition of condition A.

Syntax
```
:TRIGger:ATRigger:CONDition {ENTer|
EXIT|FALSe|TRUE}
:TRIGger:ATRigger:CONDition?
```

Example
```
:TRIGGER:ATRIGGER:CONDITION ENTER
:TRIGGER:ATRIGGER:CONDITION? ->
:TRIGGER:ATRIGGER:CONDITION ENTER
```

Description This command is valid when the trigger type is ABN or ADB.

:TRIGger:BGTime?

Function Queries all B>Time trigger settings.

Syntax
```
:TRIGger:BGTime?
```

:TRIGger:BGTime:TIME

Function Sets or queries the pulse width for B>Time triggers.

Syntax
```
:TRIGger:BGTime:TIME {<Time>}
:TRIGger:BGTime:TIME?
<Time> = 10ns to 10s
```

Example
```
:TRIGGER:BGTIME:TIME 0.001
:TRIGGER:BGTIME:TIME? -> :TRIGGER:
BGTIME:TIME 1.00000000E-03
```

:TRIGger:BLTime?

Function Queries all B<Time trigger settings.

Syntax
```
:TRIGger:BLTime?
```

:TRIGger:BLTime:TIME

Function Sets or queries the pulse width for B<Time triggers.

Syntax
```
:TRIGger:BGTime:TIME {<Time>}
:TRIGger:BGTime:TIME?
<Time> = 10ns to 10s
```

Example
```
:TRIGGER:BLTIME:TIME 0.001
:TRIGGER:BLTIME:TIME? -> :TRIGGER:
BLTIME:TIME 1.00000000E-03
```

:TRIGger:BTOut?

Function Queries all B TimeOut trigger settings.

Syntax
```
:TRIGger:BTOut?
```

:TRIGger:BTOut:TIME

Function Sets or queries the pulse width for B TimeOut triggers.

Syntax
```
:TRIGger:BTOut:TIME {<Time>}
:TRIGger:BTOut:TIME?
<Time> = 10ns to 10s
```

Example
```
:TRIGGER:BTOUT:TIME 0.001
:TRIGGER:BTOUT:TIME? -> :TRIGGER:
BTOUT:TIME 1.00000000E-03
```

:TRIGger:BBETween?

Function Queries all B Between trigger settings.

Syntax
```
:TRIGger:BBETween?
```

:TRIGger:BBETween:TIME<x>

Function Sets or queries a pulse width for B Between triggers.

Syntax
```
:TRIGger:BBETween:TIME<x> {<Time>}
:TRIGger:BBETween:TIME<x>?
<x> = 1, 2
<Time> = 10ns to 10s
```

Example
```
:TRIGger:BBET:TIME1 0.001
:TRIGGER:BBETWEEN:TIME1? -> :TRIGGER:
BBETWEEN:TIME1 1.00000000E-03
```

:TRIGger:BTRigger?

Function Queries all condition B settings.

Syntax
```
:TRIGger:BTRigger?
```

:TRIGger:BTRigger:CHANnel<x>:BIT<x>

Function Sets or queries the state of the specified bit of the specified logic channel of condition B.

Syntax
```
:TRIGger:BTRigger:CHANnel<x>:
BIT<x> {DONTcare|HIGH|LOW}
:TRIGger:BTRigger:CHANnel<x>:BIT<x>?
```

Example
```
:TRIGGER:BTRIGGER:CHANNEL9:BIT1 LOW
:TRIGGER:BTRIGGER:CHANNEL9:BIT1? ->
:TRIGGER:BTRIGGER:CHANNEL9:BIT1 LOW
```

:TRIGger:BTRigger:CHANnel<x>:SCHannel<x>:BIT<x>

Function Sets or queries the state of the specified bit of the specified sub channel of the specified logic channel of condition B.

Syntax
```
:TRIGger:BTRigger:CHANnel<x>:
SCHannel<x>:BIT<x> {DONTcare|HIGH|
LOW}
:TRIGger:BTRigger:CHANnel<x>:
SCHannel<x>:BIT<x>?
```

Example
```
:TRIGGER:BTRIGGER:CHANNEL1:
SCHANNEL1:BIT1 LOW
:TRIGGER:BTRIGGER:CHANNEL1:
SCHANNEL1:BIT1? -> :TRIGGER:BTRIGGER
:CHANNEL1:SCHANNEL1:BIT1 LOW
```

:TRIGger:BTRigger:CHANnel<x>[:CONDition]

Function Sets or queries the state of the specified channel of the specified logic channel of condition B.

Syntax
```
:TRIGger:BTRigger:
CHANnel<x>[:CONDition] {DONTcare|
HIGH|LOW}
:TRIGger:BTRigger:
CHANnel<x>[:CONDition]?
CHAN<x>'s <x> = 1 to 16
```

Example
```
:TRIGGER:BTRIGGER:CHANNEL2:
CONDITION HIGH
:TRIGGER:BTRIGGER:CHANNEL2:
CONDITION? -> :TRIGGER:BTRIGGER:
CHANNEL2:CONDITION HIGH
```

Description Use this command on channels that do not have sub channels.

:TRIGger:BTRigger:CHANnel<x>:SCHannel<x>[:CONDition]

Function Sets or queries the state of the specified sub channel of the specified logic channel of condition B.

Syntax
```
:TRIGger:BTRigger:CHANnel<x>:
SCHannel<x>[:CONDition] {DONTcare|
HIGH|LOW}
:TRIGger:BTRigger:CHANnel<x>:
SCHannel<x>[:CONDition]?
CHAN<x>'s <x> = 1 to 16
SCHannel<x>'s <x> = 1 to 16 (sub
channel)
```

Example
```
:TRIGGER:BTRIGGER:CHANNEL15:
SCHANNEL2:CONDITION HIGH
:TRIGGER:BTRIGGER:CHANNEL15:
SCHANNEL2:CONDITION? -> :TRIGGER
:BTRIGGER:CHANNEL15:SCHANNEL2:
CONDITION HIGH
```

Description Use this command on channels that have sub channels.

:TRIGger:BTRigger:CONDition

Function Sets or queries the achievement condition of condition B.

Syntax
```
:TRIGger:BTRigger:CONDition {ENTer|
EXIT}
:TRIGger:BTRigger:CONDition?
```

Example
```
:TRIGGER:BTRIGGER:CONDITION ENTER
:TRIGGER:BTRIGGER:CONDITION? ->
:TRIGGER:BTRIGGER:CONDITION ENTER
```

Description This command is valid when the trigger type is ABN or ADB.

:TRIGger:DELay

Function Sets or queries the delay.

Syntax
```
:TRIGger:DELay {<Time>}
:TRIGger:DELay?
<Time> = 0 to 10s (the resolution
varies depending on the sample rate)
(1/sample rate) × (1/10)
The minimum resolution is 10 ns.
If the sample rate is higher than 10
MS/s, the resolution will be 10 ns.
```

Example
```
:TRIGGER:DELAY 0.001
:TRIGGER:DELAY? -> :TRIGGER:
DELAY 1.0000000E-03
```

Description When the DL850/DL850V is using an external clock, the delay is fixed to 0.

:TRIGger:EOA?

Function Queries all EdgeOnA settings.

Syntax
```
:TRIGger:EOA?
```

:TRIGger:EOA:CHANnel<x>:BIT<x>

Function Sets or queries the state of the specified bit of the specified logic channel for EdgeOnA triggers.

Syntax
```
:TRIGger:EOA:CHANnel<x>:BIT<x> {OFF|
FALL|RISE}
:TRIGger:EOA:CHANnel<x>:BIT<x>?
CHAN<x>'s <x> = 1 to 16
BIT<x>'s <x> = 1 to 8
```

Example
```
:TRIGGER:EOA:CHANNEL9:BIT1 RISE
:TRIGGER:EOA:CHANNEL9:BIT1? ->
:TRIGGER:EOA:CHANNEL9:BIT1 RISE
```

:TRIGger:EOA:CHANnel<x>:SCHannel<x>:BIT<x>

Function Sets or queries the state of the specified bit of the specified sub channel of the specified logic channel for EdgeOnA triggers.

Syntax
```
:TRIGger:EOA:CHANnel<x>:SCHannel<x>:
BIT<x> {OFF|FALL|RISE}
:TRIGger:EOA:CHANnel<x>:SCHannel<x>:
BIT<x>?
CHAN<x>'s <x> = 1 to 16
BIT<x>'s <x> = 1 to 8
```

Example
```
:TRIGGER:EOA:CHANNEL15:SCHANNEL1:
CONDITION RISE
:TRIGGER:EOA:CHANNEL15:SCHANNEL1:
CONDITION? -> :TRIGGER:EOA:CHANNEL15:
SCHANNEL1:CONDITION RISE
```

:TRIGger:EOA:CHANnel<x>[:CONDition]

Function Sets or queries the state of the specified channel of the specified logic channel for EdgeOnA triggers.

Syntax
```
:TRIGger:EOA:
CHANnel<x>[:CONDition] {OFF|FALL
|RISE}
:TRIGger:EOA:CHANnel<x>[:CONDition]?
CHAN<x>'s <x> = 1 to 16
```

Example
```
:TRIGGER:EOA:CHANNEL3:CONDITION FALL
:TRIGGER:EOA:CHANNEL3:CONDITION? ->
:TRIGGER:EOA:CHANNEL3:CONDITION FALL
```

Description Use this command on channels that do not have sub channels.

:TRIGger:EOA:CHANnel<x>:SCHannel<x>[:CONDition]

Function Sets or queries the state of the specified sub channel of the specified logic channel for EdgeOnA triggers.

Syntax
```
:TRIGger:EOA:CHANnel<x>:
SCHannel[:CONDition] {OFF|FALL|RISE}
:TRIGger:EOA:CHANnel<x>:
SCHannel[:CONDition]?
CHAN<x>'s <x> = 1 to 16
SCHannel<x>'s <x> = 1 to 16 (sub
channel)
```

Example
```
:TRIGGER:EOA:CHANNEL15:SCHANNEL1:
CONDITION RISE
:TRIGGER:EOA:CHANNEL15:SCHANNEL1:
CONDITION? -> :TRIGGER:EOA:CHANNEL15:
SCHANNEL1:CONDITION RISE
```

Description Use this command on channels that have sub channels.

:TRIGger:EOA:CONDition

Function Sets or queries the achievement condition for EdgeOnA triggers.

Syntax
```
:TRIGger:EOA:CONDition {FALSe|TRUE}
:TRIGger:EOA:CONDition?
```

Example
```
:TRIGGER:EOA:CONDITION TRUE
:TRIGGER:EOA:CONDITION? -> :TRIGGER:
EOA:CONDITION TRUE
```

:TRIGger:HOLDoff?

Function Queries all hold-off settings.

Syntax
```
:TRIGger:HOLDoff?
```

:TRIGger:HOLDoff:TIME

Function Sets or queries the hold-off time.

Syntax
```
:TRIGger:HOLDoff:TIME {<Time>}
:TRIGger:HOLDoff:TIME?
<Time> = 0 to 10s (the resolution is
10 ns)
```

Example
```
:TRIGGER:HOLDOFF:TIME 0.001
:TRIGGER:HOLDOFF:TIME? -> :TRIGGER:
HOLDOFF:TIME 1.0000000E-03
```

:TRIGger:MODE

Function Sets or queries the trigger mode.

Syntax
```
:TRIGger:MODE {AUTO|ALEVel|NORMal|
SINGle|NSINgle|ONSTart}
:TRIGger:MODE?
```

Example
```
:TRIGGER:MODE AUTO
:TRIGGER:MODE? -> :TRIGGER:MODE AUTO
```

:TRIGger:OR?

Function Queries all OR trigger settings.

Syntax
```
:TRIGger:OR?
```

:TRIGger:OR:CHANnel<x>:BIT<x>

Function Sets or queries the state of the specified bit of the specified logic channel for OR triggers.

Syntax
```
:TRIGger:OR:CHANnel<x>:BIT<x> {OFF|FALL|RISE}
:TRIGger:OR:CHANnel<x>:BIT<x>?
```

Example
```
:TRIGGER:OR:CHANNEL9:BIT1 RISE
:TRIGGER:OR:CHANNEL9:BIT1? ->
:TRIGGER:OR:CHANNEL9:BIT1 RISE
```

:TRIGger:OR:CHANnel<x>:SCHannel<x>:BIT<x>

Function Sets or queries the state of the specified bit of the specified sub channel of the specified logic channel for OR triggers.

Syntax
```
:TRIGger:OR:CHANnel<x>:SCHannel<x>:BIT<x> {OFF|FALL|RISE}
:TRIGger:OR:CHANnel<x>:SCHannel<x>:BIT<x>?
```

:TRIGger:OR:CHANnel<x>[:CONDition]

Function Sets or queries the state of the specified channel of the specified logic channel for OR triggers.

Syntax
```
:TRIGger:OR:CHANnel<x>[:CONDition] {OFF|FALL|RISE|BISLope|WINIn|WINOut}
:TRIGger:OR:CHANnel<x>[:CONDition]?
```

Example
```
:TRIGGER:OR:CHANNEL1:CONDITION RISE
:TRIGGER:OR:CHANNEL1:CONDITION? ->
:TRIGGER:OR:CHANNEL1:CONDITION RISE
```

Description Use this command on channels that do not have sub channels.

:TRIGger:OR:CHANnel<x>:SCHannel<x>[:CONDition]

Function Sets or queries the state of the specified sub channel of the specified logic channel for OR triggers.

Syntax
```
:TRIGger:OR:CHANnel<x>:SCHannel<x>[:CONDition] {OFF|FALL|RISE|BISLope|WINIn|WINOut}
:TRIGger:OR:CHANnel<x>:SCHannel<x>[:CONDition]?
```

Example
```
:TRIGGER:OR:CHANNEL15:SCHANNEL1:CONDITION RISE
:TRIGGER:OR:CHANNEL15:SCHANNEL1:CONDITION? -> :TRIGGER:OR:CHANNEL15:SCHANNEL1:CONDITION RISE
```

Description Use this command on channels that have sub channels.

:TRIGger:OR:EXTernal:TYPE

Function Sets or queries the type of external trigger that is used with OR triggers.

Syntax
```
:TRIGger:OR:EXTernal:TYPE {OFF|FALL|RISE}
:TRIGger:OR:EXTernal:TYPE?
```

Example
```
:TRIGGER:OR:EXTERNAL:TYPE RISE
:TRIGGER:OR:EXTERNAL:TYPE? ->
:TRIGGER:OR:EXTERNAL:TYPE RISE
```

:TRIGger:OUT?

Function Queries all trigger output settings.

Syntax
```
:TRIGger:OUT?
```

:TRIGger:OUT:TIME

Function Sets or queries the H-pulse interval that is used when the trigger output terminal output type is 1.

Syntax
```
:TRIGger:OUT:TIME {<Time>}
:TRIGger:OUT:TIME?
<Time> = 1ms, 50ms, 100ms, 500ms,
```

Example
```
:TRIGGER:OUT:TIME 0.05
:TRIGGER:OUT:TIME? -> :TRIGGER:OUT:TIME 50.000000E-03
```

:TRIGger:OUT:TYPE

Function Sets or queries the trigger output terminal output type.

Syntax
```
:TRIGger:OUT:TYPE {<NRf>}
:TRIGger:OUT:TYPE?
<NRf> = 1, 2
```

Example
```
:TRIGGER:OUT:TYPE 2
:TRIGGER:OUT:TYPE? -> :TRIGGER:OUT:TYPE 2
```

Description The DL850/DL850V generates H pulses for a fixed interval when it triggers. This corresponds to the Pulse setting on the menu.
The DL850/DL850V generates an L pulse when it triggers. The L pulse is held during the post operation. The DL850/DL850V switches to an H pulse when the post operation finishes. This corresponds to the Normal setting on the menu.

:TRIGger:POSition

Function Sets or queries the trigger position.

Syntax
```
:TRIGger:POSition {<NRf>}
:TRIGger:POSition?
<NRf> = 0 to 100 (%, in 0.1% steps)
```

Example
```
:TRIGGER:POSITION 50
:TRIGGER:POSITION? -> :TRIGGER:POSITION 50
```

:TRIGger:SIMPle?
Function Queries all simple trigger settings.
Syntax :TRIGger:SIMPle?

:TRIGger[:SIMPle]:EXTernal:SLOPe
Function Sets or queries the external trigger slope.
Syntax :TRIGger[:SIMPle]:EXTernal:SLOPe {FALL|RISE}
:TRIGger[:SIMPle]:EXTernal:SLOPe?
Example :TRIGGER:SIMPLE:EXTERNAL:SLOPE RISE
:TRIGGER:SIMPLE:EXTERNAL:SLOPE? ->
:TRIGGER:SIMPLE:EXTERNAL:SLOPE RISE

:TRIGger[:SIMPle]:HYSTeresis
Function Sets or queries the hysteresis for simple triggers.
Syntax :TRIGger[:SIMPle]:HYSTeresis {HIGH|LOW|MIDDle}
:TRIGger[:SIMPle]:HYSTeresis?
Example :TRIGGER:SIMPLE:HYSTERESIS HIGH
:TRIGGER:SIMPLE:HYSTERESIS? ->
:TRIGGER:SIMPLE:HYSTERESIS HIGH
Description You cannot use this command to set the setting when the trigger zone is EXTernal, LINE, or TIME.

:TRIGger[:SIMPle]:LEVel
Function Sets or queries the trigger level of the channel specified by :TRIGger:SIMPle:SOURce for simple triggers .
Syntax :TRIGger[:SIMPle]:LEVel {<Voltage>|<NRf>|<Current>}
:TRIGger[:SIMPle]: LEVel?
Example :TRIGGER:SIMPLE:LEVEL 1
:TRIGGER:SIMPLE:LEVEL? -> :TRIGGER:SIMPLE:LEVEL 1.00000E+00
Description You cannot use this command to set the setting when the trigger zone is EXTernal, LINE, or TIME.
The selectable range is the range that can be measured with the current measurement range setting.
Example: If VDiv is 10, the selectable range is ± 100 V. For strain measurements, the selectable range is ±the current measurement range setting. Values do not take linear scaling information into consideration.
The trigger's selectable range of Au7Fe temperatures is 0 to 280 K (–273 to 7°C)

:TRIGger[:SIMPle]:LOGic?
Function Queries all simple trigger (logic trigger) settings.
Syntax :TRIGger[:SIMPle]:LOGic?

:TRIGger[:SIMPle]:LOGic:BIT
Function Sets or queries the bit that sets the trigger slope.
Syntax :TRIGger[:SIMPle]:LOGic:BIT {<NRf>}
:TRIGger[:SIMPle]:LOGic:BIT?
<NRf> = 1 to 8
Example :TRIGGER:SIMPLE:LOGIC:BIT 2
:TRIGGER:SIMPLE:LOGIC:BIT? ->
:TRIGGER:SIMPLE:LOGIC:BIT 2

:TRIGger[:SIMPle]:LOGic:SLOPe
Function Sets or queries the slope for logic triggers.
Syntax :TRIGger[:SIMPle]:LOGic:SLOPe {FALL|RISE}
:TRIGger[:SIMPle]:LOGic:SLOPe?
Example :TRIGGER:SIMPLE:LOGIC:SLOPE FALL
:TRIGGER:SIMPLE:LOGIC:SLOPE? ->
:TRIGGER:SIMPLE:LOGIC:SLOPE FALL

:TRIGger[:SIMPle]:SLOPe
Function Sets or queries the trigger slope of the channel specified by :TRIGger:SIMPle:SOURce for simple triggers .
Syntax :TRIGger[:SIMPle]:SLOPe { RISE|FALL|BISLope}
:TRIGger[:SIMPle]:SLOPe?
Example :TRIGGER:SIMPLE:SLOPE RISE
:TRIGGER:SIMPLE:SLOPE? -> :TRIGGER:SIMPLE:SLOPE RISE
Description You cannot use this command to set the setting when the trigger zone is EXTernal, LINE, or TIME.

:TRIGger:SIMPle:SOURce
Function Sets or queries the source for simple triggers.
Syntax :TRIGger:SIMPle:SOURce {<NRf>[,<NRf>]|EXTernal|LINE|TIME}
:TRIGger:SIMPle:SOURce?
<NRf> = 1 to 16
[,<NRf>] = If the channel has sub channels, this is the range of sub channel numbers.
Example :TRIGGER:SIMPLE:SOURCE 1
:TRIGGER:SIMPLE:SOURCE? -> :TRIGGER:SIMPLE:SOURCE 1

:TRIGger:SOURce?
Function Queries all the settings of the trigger source for enhanced triggers.
Syntax :TRIGger:SOURce?

:TRIGger:SOURce:CHANnel<x>[:SCHannel<x>]?

Function Queries all the settings of the specified channel's trigger source for enhanced triggers.

Syntax
```
:TRIGger:SOURce:
CHANnel<x>[:SCHannel<x>]?
```

:TRIGger:SOURce:CHANnel<x>[:SCHannel<x>]:CENTer

Function Sets or queries the window trigger center for enhanced triggers.

Syntax
```
:TRIGger:SOURce:
CHANnel<x>[:SCHannel<x>]:
CENTer {<Voltage>|<NRf>|<Current>}
:TRIGger:SOURce:CHANnel<x>[:
SCHannel<x>]:CENTer?
CHANnel<x>'s <x> = 1 to 16
SCHannel<x>'s <x> = 1 to 16
```

Example
```
:TRIGGER:SOURCE:CHANNEL15:SCHANNEL1:
CENTER 1
:TRIGGER:SOURCE:CHANNEL15:SCHANNEL1:
CENTER? -> :TRIGGER:SOURCE:CHANNEL15:
SCHANNEL1:CENTER 1.00000E+00
```

:TRIGger:SOURce:CHANnel<x>[:SCHannel<x>]:HYSTeresis

Function Sets or queries the specified channel's hysteresis for enhanced triggers.

Syntax
```
:TRIGger:SOURce:
CHANnel<x>[:SCHannel<x>]:
HYSTeresis {HIGH|LOW|MIDDle}
CHANnel<x>'s <x> = 1 to 16
SCHannel<x>'s <x> = 1 to 16
```

Example
```
:TRIGGER:SOURCE:CHANNEL15:SCHANNEL1:
HYSTERESIS HIGH
:TRIGGER:SOURCE:CHANNEL15:SCHANNEL1:
HYSTERESIS? -> :TRIGGER:SOURCE:
CHANNEL15:SCHANNEL1:HYSTERESIS HIGH
```

:TRIGger:SOURce:CHANnel<x>[:SCHannel<x>]:LEVel

Function Sets or queries the specified channel's level for enhanced triggers.

Syntax
```
:TRIGger:SOURce:
CHANnel<x>[:SCHannel<x>]:
LEVel{<Voltage>|<NRf>|<Current>}
:TRIGger:SOURce:CHANnel<x>[:
SCHannel<x>]:LEVel?
CHANnel<x>'s <x> = 1 to 16
SCHannel<x>'s <x> = 1 to 16
```

Example
```
:TRIGGER:SOURCE:CHANNEL15:SCHANNEL1:
LEVEL 1
:TRIGGER:SOURCE:CHANNEL15:SCHANNEL1:
LEVEL? -> :TRIGGER:SOURCE:CHANNEL15:
SCHANNEL1:LEVEL 1
```

:TRIGger:SOURce:CHANnel<x>[:SCHannel<x>]:WIDTh

Function Sets or queries the window trigger width for enhanced triggers.

Syntax
```
:TRIGger:SOURce:
CHANnel<x>[:SCHannel<x>]:
WIDTh {<Voltage>|<NRf>|<Current>}
:TRIGger:SOURce:
CHANnel<x>[:SCHannel<x>]:WIDTh?
CHANnel<x>'s <x> = 1 to 16
SCHannel<x>'s <x> = 1 to 16
```

Example
```
:TRIGGER:SOURCE:CHANNEL15:SCHANNEL1:
WIDTH 2
:TRIGGER:SOURCE:CHANNEL15:SCHANNEL1:
WIDTH? -> :TRIGGER:SOURCE:CHANNEL15:
SCHANNEL1:WIDTH 2
```

:TRIGger:SOURce:CHANnel<x>:WWIDth

Function Sets or queries the width for wave window triggers.

Syntax
```
:TRIGger:SOURce:CHANnel<x>:
WWIDth {<Voltage>|<NRf>|<Current>}
:TRIGger:SOURce:CHANnel<x>:WWIDth?
CHANnel<x>'s <x> = 1 to 16
```

Example
```
:TRIGGER:SOURCE:CHANNEL1:WWIDTH 0.5
:TRIGGER:SOURCE:CHANNEL1:WWIDTH? ->
:TRIGGER:SOURCE:CHANNEL1:WWIDTH
```

:TRIGger:TGTime?

Function Queries all T>Time trigger settings.

Syntax
```
:TRIGger:TGTime?
```

:TRIGger:TGTime:TIME

Function Sets or queries the pulse width for T>Time triggers.

Syntax
```
:TRIGger:TGTime:TIME {<Time>}
:TRIGger:TGTime:TIME?
<Time> = 10ns to 10s
```

Example
```
:TRIGGER:TGTIME:TIME 0.001
:TRIGGER:TGTIME:TIME? -> :TRIGGER
:TGTIME:TIME 1.00000000E-03
```

:TRIGger:TIMer?

Function Queries all time trigger settings.

Syntax
```
:TRIGger:TIMer?
```

:TRIGger:TIMer:DATE

Function Sets or queries the date for time triggers.

Syntax
```
:TRIGger:TIMer:DATE <String>
:TRIGger:TIMer:DATE?
<String> = YYYY/MM/DD
YYYY: 2000 to 2099
```

Example
```
:TRIGGER:TIMER:DATE "2010/01/01"
:TRIGGER:TIMER:DATE? -> :TRIGGER:
TIMER:DATE "2010/01/01"
```

:TRIGger:TIMer:INTerval

Function Sets or queries the trigger interval for time triggers.

Syntax
```
:TRIGger:TIMer:INTerval {SEC10|SEC15
|SEC20|SEC30|SEC40|SEC50|MIN1|MIN2|
MIN3|MIN4|MIN5|MIN6|MIN7|MIN8|MIN9|
MIN10|MIN15|MIN20|MIN25|MIN30|MIN40|
MIN45|MIN50|HOUR1|HOUR2|HOUR3|HOUR4|
HOUR5|HOUR6|HOUR7|HOUR8|HOUR9|HOUR10|
HOUR11|HOUR12|HOUR18|HOUR24}
TRIGger:TIMer:INTerval?
```

Example
```
:TRIGGER:TIMER:INTERVAL HOUR1
:TRIGGER:TIMER:INTERVAL? ->
:TRIGGER:TIMER:INTERVAL HOUR1
```

:TRIGger:TIMer:TIME

Function Sets or queries the time for time triggers.

Syntax
```
:TRIGger:TIMer:TIME <String>
:TRIGger:TIMer:TIME?
<String> = HH:MM:SS
```

Example
```
:TRIGGER:TIMER:TIME "00:00:00"
:TRIGGER:TIMER:TIME? -> :TRIGGER:
TIMER:TIME "00:00:00"
```

:TRIGger:TITime?

Function Queries all T1<T<T2 trigger settings.

Syntax
```
:TRIGger:TITime?
```

:TRIGger:TITime:TIME<x>

Function Sets or queries the pulse width for T1<T<T2 triggers.

Syntax
```
:TRIGger:TITime:TIME<x> {<Time>}
:TRIGger:TITime:TIME<x>?
<Time> = 10ns to 10s
<x> = 1, 2
```

Example
```
:TRIGGER:TITIME:TIME1 0.001
:TRIGGER:TITIME:TIME1? -> :TRIGGER:
TITIME:TIME1 1.00000000E-03
```

:TRIGger:TLTime?

Function Queries all T<Time trigger settings.

Syntax
```
:TRIGger:TLTime?
```

:TRIGger:TLTime:TIME

Function Sets or queries the pulse width for T<Time triggers.

Syntax
```
:TRIGger:TLTime:TIME {<Time>}
:TRIGger:TLTime:TIME?
<Time> = 10ns to 10s
```

Example
```
:TRIGGER:TLTIME:TIME 0.001
:TRIGGER:TLTIME:TIME? -> :TRIGGER:
TLTIME:TIME 1.00000000E-03
```

:TRIGger:TOTime?

Function Queries all T<T1,T2<T trigger settings.

Syntax
```
:TRIGger:TOTime?
```

:TRIGger:TOTime:TIME<x>

Function Sets or queries the pulse width for T<T1,T2<T triggers.

Syntax
```
:TRIGger:TOTime:TIME<x> {<Time>}
:TRIGger:TOTime:TIME<x>?
<Time> = 10ns to 10s
<x> = 1, 2
```

Example
```
:TRIGGER:TOTIME:TIME1 0.001
:TRIGGER:TOTIME:TIME1? -> :TRIGGER:
TOTIME:TIME1 1.00000000E-03
```

:TRIGger:TYPE

Function Sets or queries the trigger type.

Syntax
```
:TRIGger:TYPE {ABN|ADB|AND|EOA|
BBETween|BLTime|BGTime|BTOut|OR|
SIMPle|TGTime|TLTime|TITime|TOTime|
WWINdow}
:TRIGger:TYPE?
```

Example
```
:TRIGGER:TYPE SIMPLE
:TRIGGER:TYPE? -> :TRIGGER:
TYPE SIMPLE
```

:TRIGger:WWINdow?

Function Queries all wave window trigger settings.

Syntax
```
:TRIGger:WWINdow?
```

:TRIGger:WWINdow:CHANnel<x>

Function Sets or queries the state of the specified channel for wave window triggers.

Syntax
```
:TRIGger:WWINdow:
CHANnel<x> {<Boolean>}
:TRIGger:WWINdow:CHANnel<x>?
<x> = 1 to 16
```

Example
```
:TRIGGER:WWINDOW:CHANNEL1 ON
:TRIGGER:WWINDOW:CHANNEL1? ->
:TRIGGER:WWINDOW:CHANNEL1 1
```

Description The DL850/DL850V cannot perform wave window triggering on modules that have sub channels, logic modules, modules that are performing temperature measurements, and frequency modules.

:TRIGger:WWINdow:FREQuency

Function Sets or queries the cycle frequency for wave window triggers.

Syntax
```
:TRIGger:WWINdow:
FREQuency {<Frequency>}
:TRIGger:WWINdow:FREQuency?
<Frequency> = 40 to 1000 (Hz)
```

Example
```
:TRIGGER:WWINDOW:FREQUENCY 400
:TRIGGER:WWINDOW:FREQUENCY? ->
:TRIGGER:WWINDOW:FREQUENCY 400
```

:TRIGger:WWINdow:REFCycle

Function Sets or queries the reference cycle for wave window triggers.

Syntax
```
:TRIGger:WWINdow:REFCycle {<NRf>}
:TRIGger:WWINdow:REFCycle?
<NRf> = 1, 2, 4
```

Example
```
:TRIGGER:WWINDOW:REFCYCLE 1
:TRIGGER:WWINDOW:REFCYCLE? ->
:TRIGGER:WWINDOW:REFCYCLE 1
```

:TRIGger:WWINdow:SYNC?

Function Queries all synchronization channel settings for wave window triggers.

Syntax
```
:TRIGger:WWINdow:SYNC?
```

:TRIGger:WWINdow:SYNC:HYSTeresis

Function Sets or queries the hysteresis of the synchronization channel for wave window triggers.

Syntax
```
:TRIGger:WWINdow:SYNC:
HYSTeresis {HIGH|LOW|MIDDle}
:TRIGger:WWINdow:SYNC:HYSTeresis?
```

Example
```
:TRIGGER:WWINDOW:SYNC:
HYSTERESIS MIDDLE
:TRIGGER:WWINDOW:SYNC:HYSTERESIS? ->
:TRIGGER:WWINDOW:SYNC:
HYSTERESIS MIDDLE
```

Description You cannot set this setting when the synchronization channel is set to AUTO.

:TRIGger:WWINdow:SYNC:LEVel

Function Sets or queries the level of the synchronization channel for wave window triggers.

Syntax
```
:TRIGger:WWINdow:SYNC:LEVel
{<Voltage>|
<NRf>|<Current>}
:TRIGger:WWINdow:SYNC:LEVel?
```

Example
```
:TRIGGER:WWINDOW:SYNC:
LEVEL 2.00000E+00
:TRIGGER:WWINDOW:SYNC:LEVEL? -> :
TRIGGER:WWINDOW:SYNC:
LEVEL 2.00000E+00
```

Description You cannot set this setting when the synchronization channel is set to AUTO.

:TRIGger:WWINdow:SYNC:TRACe

Function Sets or queries the synchronization channel for wave window triggers.

Syntax
```
:TRIGger:WWINdow:SYNC:TRACe {<NRf>|
AUTO}
:TRIGger:WWINdow:SYNC:TRACe?
<NRf> = 1 to 16
```

Example
```
:TRIGGER:WWINDOW:SYNC:TRACE 1
:TRIGGER:WWINDOW:SYNC:TRACE? ->
:TRIGGER:WWINDOW:SYNC:TRACE 1
```

5.35 WAVeform Group

The commands in this group deal with the acquired waveform data. There are no front panel keys that correspond to the commands in this group.

:WAVeform?

Function Queries all waveform data output settings.
Syntax :WAVeform?

:WAVeform:BITS?

Function Queries the bit length of the waveform data specified by the :WAVeform:TRACe command.
Syntax :WAVeform:BITS?
Example :WAVEFORM:BITS? -> :WAVEFORM:BITS 16

:WAVeform:BYTeorder

Function Sets or queries the transmission byte order for data formats that are 2 bytes or longer.
Syntax :WAVeform:BYTeorder {LSBFirst|MSBFirst}
:WAVeform:BYTeorder?
Example :WAVEFORM:BYTEORDER LSBFIRST
:WAVEFORM:BYTEORDER? -> :WAVEFORM:BYTEORDER LSBFIRST
Description This command is only valid in the :WAVeform group.

:WAVeform:CAPTure?

Function Queries all capture data output settings.
Syntax :WAVeform:CAPTure?

:WAVeform:CAPTure:END

Function Sets or queries the end point to use when capturing waveform data.
Syntax :WAVeform:CAPTure:END {<NRf>}
:WAVeform:CAPTure:END?
<NRf> = 0 to (the number of data points in the waveform that is being captured − 1)
Example :WAVEFORM:CAPTURE:END 10000
:WAVEFORM:CAPTURE:END? -> :WAVEFORM:CAPTURE:END 10000

:WAVeform:CAPTure:LENGth?

Function Queries the number of data points in the captured waveform.
Syntax :WAVeform:CAPTure:LENGth?
Example :WAVEFORM:CAPTURE:LENGTH? ->
:WAVEFORM:CAPTURE:LENGTH 10010

:WAVeform:CAPTure:RECord

Function Sets or queries the record number to use when capturing waveform data.
Syntax :WAVeform:CAPTure:RECord {MAXimum|MINimum|<NRf>}
:WAVeform:CAPTure:RECord?
<NRf> = −5000 to 5000
Example :WAVEFORM:CAPTURE:RECORD -1
:WAVEFORM:CAPTURE:RECORD? ->
:WAVEFORM:CAPTURE:RECORD -1

:WAVeform:CAPTure:RECord? MAXimum

Function Queries the largest record number of the captured waveform.
Syntax :WAVeform:CAPTure:RECord? MAXimum
Example :WAVEFORM:CAPTURE:RECORD? MAXIMUM ->
:WAVEFORM:CAPTURE:RECORD 0

:WAVeform:CAPTure:RECord? MINimum

Function Queries the smallest record number of the captured waveform.
Syntax :WAVeform:CAPTure:RECord? MINimum
Example :WAVEFORM:CAPTURE:RECORD? MINIMUM ->
:WAVEFORM:CAPTURE:RECORD -1

:WAVeform:CAPTure:SEND?

Function Queries specified capture waveform data.
Syntax :WAVeform:CAPTure:SEND? {<NRf>}
<NRf> = 1 to 5000
Example :WAVeform:CAPTure:SEND?
Description <NRf> can be omitted. If you specify <NRf>, waveform data is queried <NRf> times in order starting from the record located at the record number specified by :WAVeform:CAPTure:RECord – <NRf>.
An error will occur if the waveform that WAVeform commands will be applied to is set as an event.

:WAVeform:CAPTure:SRATe?

Function Queries the sample rate of the capture waveform specified by the :WAVeform:TRACe command.

Syntax `:WAVeform:CAPTure:SRATe?`

Example
```
:WAVEFORM:CAPTURE:SRATE 1000
:WAVEFORM:CAPTURE:SRATE? ->
:WAVEFORM:CAPTURE:
SRATE 1.0000000E+03
```

Description This command returns the sample rate of measured data.
An error will occur if the waveform that WAVeform commands will be applied to is set as an event.

:WAVeform:CAPTure:STARt

Function Sets or queries the start point to use when capturing waveform data.

Syntax
```
:WAVeform:CAPTure:STARt {<NRf>}
:WAVeform:CAPTure:STARt?
```
`<NRf> = 0 to (the number of acquired points in the captured waveform - 1)`

Example
```
:WAVEFORM:CAPTURE:START 0
:WAVEFORM:CAPTURE:START? ->
:WAVEFORM:CAPTURE:START 0
```

:WAVeform:CAPTure:TRIGger?

Function Queries the trigger position (represented as a number of points) in the captured waveform.

Syntax `:WAVeform:CAPTure:TRIGger?`

Example
```
:WAVEFORM:CAPTURE:TRIGGER? ->
:WAVEFORM:CAPTURE:TRIGGER 5000
```

:WAVeform:DATaselect

Function Queries whether to query the waveform specified by the :WAVeform:TRACe command using ACQ data or PP data.

Syntax
```
:WAVeform:DATaselect {ACQData|
PPData}
:WAVeform:DATaselect?
```

Example
```
:WAVEFORM:DATASELECT ACQDATA
:WAVEFORM:DATASELECT? -> :WAVEFORM:
DATASELECT ACQDATA
```

:WAVeform:END

Function Sets or queries the end data point in the waveform specified by the :WAVeform:TRACe command (the main waveform).

Syntax
```
:WAVeform:END {<NRf>}
:WAVeform:END?
```
`<NRf> = 0 to 50100000000`

Example
```
:WAVEFORM:END 10000
:WAVEFORM:END? -> :WAVEFORM:
END 10000
```

Description You can query the total number of data points that have been acquired by using the :WAVeform:LENGth? command.

:WAVeform:FORMat

Function Sets or queries the transmission data format.

Syntax
```
:WAVeform:FORMat
{ASCii|BYTE|WORD|DWORd}
:WAVeform:FORMat?
```

Example
```
:WAVEFORM:FORMAT ASCII
:WAVEFORM:FORMAT? -> :WAVEFORM:
FORMAT ASCII
```

Description The "DWORd" format setting is valid on a CAN bus monitor module when the bit length of the transmission data is greater than 17.

:WAVeform:LENGth?

Function Queries the total number of data points in the waveform specified by the :WAVeform:TRACe command (the main waveform).

Syntax `:WAVeform:LENGth?`

Example
```
:WAVEFORM:LENGTH? -> :WAVEFORM:
LENGTH 10010
```

Description The total number of data points varies depending on the DL850/DL850V settings. For details, see appendix 1, “Relationship between the Time Axis Setting, Record Length, and Sample Rate” in the *DL850/DL850V Startup Guide*.

:WAVeform:MODule?

Function Queries the module of the waveform specified by the :WAVeform:TRACe command.

Syntax `:WAVeform:MODule?`

Example
```
:WAVEFORM:MODULE? -> :WAVEFORM:
MODULE M701250
```

Description The values returned for each module are listed below.

NOMODULE	No module
M701250	701250 (HS10M12)
M701251	701251 (HS1M16)
M701255	701255 (NONISO_10M12)
M701260	701260 (HV (with RMS))
M701261	701261 (Voltage/Temp.)
M701262	701262 (Voltage/Temp. with AAF)
M701265	701265 (TEMP/HPV)
M701270	701270 (STRAIN_NDIS)
M701271	701271 (STRAIN_DSUB)
M701275	701275 (ACCL/VOLT)
M701280	701280 (FREQ)
M720210	720210 (HS100M12)
M720220	720220 (16CH VOLT)
M720230	720230 (LOGIC)
M720240	720240 (CAN)

If the waveform that WAVeform commands will be applied to is set as an event, “EVENT” is returned.

:WAVeform:OFFSet?

Function Queries the offset value used to convert the waveform data specified by the :WAVeform:TRACe command to physical values.

Syntax `:WAVeform:OFFSet?`

Example `:WAVEFORM:OFFSET? -> :WAVEFORM:OFFSET 0.0000000E+00`

Description When you are using linear scaling, this command returns a value that includes the scaling data. If the source waveform is logic data, this command returns 0.

:WAVeform:RANGe?

Function Queries the measurement range used to convert the waveform data specified by the :WAVeform:TRACe command to physical values.

Syntax `:WAVeform:RANGe?`

Example `:WAVEFORM:RANGE? -> :WAVEFORM:RANGE 50.000000E+00`

Description When you are using linear scaling, this command returns a value that includes the scaling data. If the source waveform is logic data, this command returns 1.

:WAVeform:RECord

Function Sets or queries the main waveform record number that WAVeform commands will be applied to.

Syntax `:WAVeform:RECord {MINimum|<NRf>}`
`:WAVeform:RECord?`
`<NRf> = 0 to -4999`

Example `:WAVEFORM:RECORD -4`
`:WAVEFORM:RECORD? -> :WAVEFORM:RECORD -4`

Description Specify “MINimum” to specify the minimum record number.
The minimum record number is the record number of the oldest data.

:WAVeform:RECord? MINimum

Function Queries the minimum record number in the history memory (for the main waveform).

Syntax `:WAVeform:RECord? MINimum`

Example `:WAVEFORM:RECORD? -> :WAVEFORM:RECORD -8`

:WAVeform:SEND?

Function Queries the waveform data specified by the :WAVeform:TRACe command (the main waveform data).

Syntax `:WAVeform:SEND? {<NRf>}`
`<NRf> = 1 to 5000 (this varies depending on the measurement time, sample rate, and memory model)`

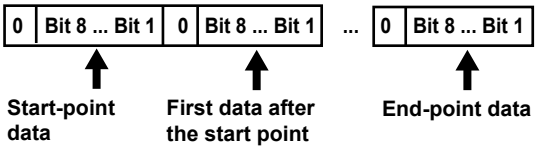
Example
- When data is being sent in BYTE or WORD format
  `:WAVeform:SEND? -> #9? (9-digit number of bytes; data byte sequence)`
- When data is being sent in ASCII format
  `:WAVeform:SEND? -> <NRf>,<NRf>,<NRf>...`

Description <NRf> can be omitted. If you specify <NRf>, waveform data is queried <NRf> times in order starting from the record located at the record number specified by :WAVeform:RECord – <NRf>.
If the number of bytes of binary data that are being sent exceeds the 9-digit number, the data will not be sent.
If the output format is PP and the original data is less than 10 div, only the queried portion of the data is sent.
When the output format is PP, the output start and end points are disabled.
The output format for logic and event waveforms is shown below.

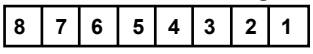
When Data Transmitted in BYTE Format

When Data Transmitted in WORD (LSBF) Format

When Data Transmitted in WORD (MSBF) Format



Event Waveform Bit Assignments



Bit 1: Dual capture event
Bit 2: Manual event
Bits 3 to 8: Reserved

:WAVeform:SIGN?

Function Queries whether signs are included in the block data of the source waveform data specified by : WAVeform:TRACe when the data is queried.

Syntax `:WAVeform:SIGN?`

Example `:WAVEFORM:SIGN? -> :WAVEFORM:SIGN 1`

Description This command returns 0 if the source waveform is a logic waveform and 1 otherwise.

:WAVeform:SRATe? (Sample RATE)

Function Queries the sample rate of the waveform specified by the :WAVeform:TRACe command.

Syntax `:WAVeform:SRATe?`

Example
```
:WAVEFORM:SRATE? -> :WAVEFORM:
SRATE 500.0E+03
```

Description This command returns the sample rate of measured data.

:WAVeform:STARt

Function Sets or queries the start data point in the waveform specified by the :WAVeform:TRACe command (the main waveform).

Syntax
```
:WAVeform:STARt {<NRf>}
:WAVeform:STARt?
<NRf> = 0 to (the number of data
points - 1)
```

Example
```
:WAVEFORM:START 0
:WAVEFORM:START? -> :WAVEFORM:
START 0
```

Description You can query the total number of data points that have been acquired by using the :WAVeform: LENGth? command.

:WAVeform:TRACe

Function Sets or queries the waveform that WAVeform commands will be applied to.

Syntax
```
:WAVeform:TRACe {EVENt|MATH<x>|
<NRf> [,<NRf>]}
:WAVeform:TRACe?
<NRf> = 1 to 16
<NRf> = 1 to 16      (sub channel)
MATH<x>'s <x> = 1 to 8
EVENt: Sends event data
```

Example
```
:WAVEFORM:TRACE 1
:WAVEFORM:TRACE? -> :WAVEFORM:
TRACE 1
```

Description An error will occur if nothing is installed in the specified channel.

:WAVeform:TRIGger?

Function Queries the trigger position of the record specified by the :WAVeform:RECord command.

Syntax `:WAVeform:TRIGger?`

Example
```
:WAVEFORM:TRIGGER? -> :WAVEFORM:
TRIGGER 5000
```

Description Queries the number of points from the beginning of the record to the trigger position.

:WAVeform:TYPE?

Function Queries the acquisition mode of the source waveform.

Syntax `:WAVeform:TYPE?`

Example
```
:WAVEFORM:TYPE? -> :WAVEFORM:
TYPE NORMAL
```

Description This command returns “AVERage,” “ENVelope,” “BAVerage,” or “NORMal.”

5.36 XY Group

The commands in this group deal with the XY display. You can make the same settings and queries that you can by using the SHIFT+DISPLAY (X-Y) keys on the front panel.

:XY?

Function Queries all XY settings.

Syntax
```
:XY?
```

:XY:CDISplay (Combine Display)

Function Sets or queries whether to combine the displays of Window1 and Window2 on the XY waveform display.

Syntax
```
:XY:CDISplay {<Boolean>}
:XY:CDISplay?
```

Example
```
:XY:CDISPLAY ON
:XY:CDISPLAY? -> :XY:CDISPLAY 1
```

:XY:DOTConnect

Function Sets or queries whether dot connect is ON or OFF for XY waveforms.

Syntax
```
:XY:DOTConnect {<Boolean>}
:XY:DOTConnect?
```

Example
```
:XY:DOTCONNECT ON
:XY:DOTCONNECT? -> :XY:DOTCONNECT 1
```

:XY:DECimation

Function Sets or queries the number of dots (after decimation) that XY waveforms use.

Syntax
```
:XY:DECimation {<NRf>}
:XY:DECimation?
<NRf>=2000, 100000
```

Example
```
:XY:DECIMATION 2000
:XY:DECIMATION? -> :XY:
DECIMATION 2000
```

:XY:MARKer

Function Sets or queries whether XY pen markers are on.

Syntax
```
:XY:MARKer {<Boolean>}
:XY:MARKer?
```

Example
```
:XY:MARKER ON
:XY:MARKER? -> :XY:MARKER 1
```

:XY:TCLear (Trace Clear On Start)

Function Sets or queries whether the XY trace-clear-on-start feature is on.

Syntax
```
:XY:TCLear {<Boolean>}
:XY:TCLear?
```

Example
```
:XY:TCLEAR ON
:XY:TCLEAR? -> :XY:TCLEAR 1
```

:XY:WINDow<x>?

Function Queries all settings related to the specified XY window.

Syntax
```
:XY:WINDow<x>?
```

:XY:WINDow<x>:TRANge

Function Sets or queries the T-Y waveform range to display in the XY window.

Syntax
```
:XY:WINDow<x>:TRANge {<NRf>,<NRf>}
:XY:WINDow<x>:TRANge?
<NRf> = -5.00 to 5.00div
Resolution: 10/record length
```

Example
```
:XY:WINDOW1:TRANGE -5,5
:XY:WINDOW1:TRANGE? -> :XY:
WINDOW1:TRANGE -5.000000000000,
5.000000000000
```

:XY:WINDow<x>:MODE

Function Sets or queries whether the specified XY window is displayed.

Syntax
```
:XY:WINDow<x>:MODE {<Boolean>}
:XY:WINDow<x>:MODE?
<x> = 1, 2
```

Example
```
:XY:WINDOW1:MODE ON
:XY:WINDOW1:MODE? -> :XY:WINDOW1:
MODE 1
```

:XY:WAVeform<x1>?

Function Queries all settings related to the specified XY waveform.

Syntax
```
:XY:WAVeform<x1>?
```

:XY:WAVeform<x1>:DISPlay

Function Sets or queries whether the specified XY waveform is displayed.

Syntax
```
:XY:WAVeform<x1>:DISPlay {<Boolean>}
:XY:WAVeform<x1>:DISPlay?
<x1> = 1 to 8
```

Example
```
:XY:WAVEFORM1:DISPLAY ON
:XY:WAVEFORM1:DISPLAY? -> :XY:
WAVEFORM1:DISPLAY 1
```

:XY:WAVeform<x1>:XTRace

Function Sets or queries the channel that is assigned to the specified XY waveform's X axis.

Syntax
```
:XY:WAVeform<x1>:
XTRace {<NRf> [,<NRf>]|MATH<x2>}
:XY:WAVeform<x1>:XTRace?
<x1> = 1 to 8
<NRf> = 1 to 16
<NRf> = 1 to 16      (sub channel)
<x2> = 1 to 8
```

Example
```
:XY:WAVEFORM1:XTRACE 1
:XY:WAVEFORM1:XTRACE? -> :XY:
WAVEFORM1:XTRACE 1
```

Description You cannot select the channel of a logic module.

:XY:WAVeform<x1>:YTRace

Function Sets or queries the channel that is assigned to the specified XY waveform's Y axis.

Syntax
```
:XY:WAVeform<x1>:
YTRace {<NRf>[,<NRf>]|MATH<x2>}
:XY:WAVeform<x1>:YTRace?
<x1> = 1 to 8
<NRf> = 1 to 16
<NRf> = 1 to 16      (sub channel)
<x2> = 1 to 8
```

Example
```
:XY:WAVEFORM1:YTRACE 2
:XY:WAVEFORM1:YTRACE? -> :XY:
WAVEFORM1:YTRACE 2
```

Description You cannot select the channel of a logic module.

5.37 ZOOM Group

The commands in this group deal with the zoomed display. You can make the same settings and queries that you can by using the ZOOM key on the front panel.
If the selectable range of the time axis is "<NRf> = –5 to 5div," the selectable range varies depending on settings such as the record length. For details, see "Selectable Range of Cursor Positions" under "Notes about Cursor Measurement" in chapter 9 of IM DL850-01EN.

:ZOOM?
Function Queries all zoom settings.
Syntax :ZOOM?

:ZOOM:ASCRoll?
Function Queries all auto scroll settings.
Syntax :ZOOM:ASCRoll?

:ZOOM:ASCRoll:JUMP
Function Moves the center position of the zoom box to the left or right edge of the main window.
Syntax :ZOOM:ASCRoll:JUMP {LEFT|RIGHt}
Example :ZOOM:ASCROLL:JUMP RIGHT

:ZOOM:ASCRoll:SPEed
Function Sets or queries the auto scroll speed of the zoom box.
Syntax :ZOOM:ASCRoll:SPEed {<NRf>}
:ZOOM:ASCRoll:SPEed?
<NRf> = 1 to 10
Example :ZOOM:ASCROLL:SPEED 5
:ZOOM:ASCROLL:SPEED? -> :ZOOM:ASCROLL:SPEED 5

:ZOOM:ASCRoll:STARt
Function Starts auto scrolling.
Syntax :ZOOM:ASCRoll:STARt {LEFT|RIGHt}
Example :ZOOM:ASCROLL:STAR LEFT

:ZOOM:ASCRoll:STOP
Function Stops auto scrolling.
Syntax :ZOOM:ASCRoll:STOP
Example :ZOOM:ASCROLL:STOP

:ZOOM:ASCRoll:TARGet
Function Sets or queries the zoom window that will be auto scrolled.
Syntax :ZOOM:ASCRoll:TARGet {Z1|Z2}
:ZOOM:ASCRoll:TARGet?
Example :ZOOM:ASCROLL:TARGET Z1
:ZOOM:ASCROLL:TARGET? -> :ZOOM:ASCROLL:TARGET Z1

:ZOOM:FORMat<x>
Function Sets or queries the display format of the specified zoom waveform.
Syntax :ZOOM:FORMat<x> {MAIN|<NRf>}
:ZOOM:FORMat<x>?
<NRf> = 1, 2, 3, 4, 6, 8, 12, 16
<x> = 1, 2 (sets the target zoom box)
Example :ZOOM:FORMAT1 1
:ZOOM:FORMAT1? -> :ZOOM:FORMAT1 1

:ZOOM:GROup<x1>?
Function Queries all settings related to the specified group display of zoom waveforms.
Syntax :ZOOM:GROup<x1>?

:ZOOM:GROup<x1>:TRACe<x2>
Function Sets or queries whether the specified source waveform of the specified group display of zoom waveforms is displayed.
Syntax :ZOOM:GROup<x1>:TRACe<x2> {<Boolean>}
:ZOOM:GROup<x1>:TRACe<x2>?
<x1> = 1 to 4 (sets the group display)
<x2> = 1 to 64 (sets the source waveform)
Example :ZOOM:GROUP1:TRACE1 1
:ZOOM:GROUP1:TRACE1? -> :ZOOM:GROUP1:TRACE1 1
Description ZOOM1 and ZOOM2 both reference the same setting. Any changes made to the dual capture setting do not affect this setting, nor do any changes made to this setting affect the dual capture setting.

:ZOOM:MAG<x>

Function Sets or queries the horizontal magnification of the specified zoom waveform.

Syntax
```
:ZOOM:MAG<x> {<NRf>}
:ZOOM:MAG<x>?
<x> = 1, 2 (sets the target zoom box)
<NRf> = 1 to 200000000
```

Example
```
:ZOOM:MAG1 100
:ZOOM:MAG1? -> :ZOOM:MAG1 100.0
```

Description Use this command when the DL850/DL850V is using an external clock. When the DL850/DL850V is using the internal clock, use the :ZOOM:TDIV<x> command to set the magnification. The magnification is set as a percentage.

:ZOOM:MAIN

Function Sets or queries the proportion of the main waveform display area that is used when zooming waveforms.

Syntax
```
:ZOOM:MAIN {20|50|OFF}
:ZOOM:MAIN?
```

Example
```
:ZOOM:MAIN 20
:ZOOM:MAIN? -> :ZOOM:MAIN 20
```

:ZOOM:MODE<x>

Function Sets or queries whether the specified zoom waveform is displayed.

Syntax
```
:ZOOM:MODE<x> {<Boolean>}
:ZOOM:MODE<x>?
```

Example
```
:ZOOM:MODE1 ON
:ZOOM:MODE1? -> :ZOOM:MODE1 1
```

:ZOOM:MOVE

Function Sets or queries whether the movement of the zoom position to the latest position is enabled or disabled.

Syntax
```
:ZOOM:MOVE<x> {<Boolean>}
:ZOOM:MOVE<x>?
```

Example
```
:ZOOM:MOVE1 ON
:ZOOM:MODE1? -> :ZOOM:MODE1 1
```

:ZOOM:POSition<x>

Function Sets or queries the zoom position of the specified zoom waveform.

Syntax
```
:ZOOM:POSition<x> {<NRf>}
:ZOOM:POSition<x>?
<x> = 1, 2 (sets the target zoom box)
<NRf> = -5 to 5 div
```

Example
```
:ZOOM:POS1 2
:ZOOM:POS1? -> :ZOOM:POS1 2.000000000000
```

:ZOOM:TDIV<x>

Function Sets or queries the T/DIV value of the specified zoom waveform.

Syntax
```
:ZOOM:TDIV<x> {<Time>|HOUR1|HOUR2|HOUR3|HOUR4|HOUR5|HOUR6|HOUR7|HOUR8|HOUR9|HOUR10|HOUR12|DAY1|DAY2|DAY3}
:ZOOM:TDIV<x>?
<x> = 1, 2 (sets the target zoom box)
<Time> = the same as the T/Div setting
```

Example
```
:ZOOM:TDIV1 100us
:ZOOM:TDIV1? -> :ZOOM:TDIV1 100.0E-06
```

Description Use this command when the DL850/DL850V is using the internal clock. When the DL850/DL850V is using an external clock, use the :ZOOM:MAG<x> command to set the magnification.

:ZOOM:WLAYout (Window Layout)

Function Sets or queries the window layout that is used when waveforms are zoomed.

Syntax
```
:ZOOM:WLAYout {VERTical|SIDE}
:ZOOM:WLAYout?
```

Example
```
:ZOOM:WLAY SIDE
:ZOOM:WLAY? -> :ZOOM:WLAY SIDE
```

:ZOOM:Z2Target

Function Sets or queries the source window of Z2 when both Z1 and Z2 are displayed.

Syntax
```
:ZOOM:Z2Target {MAIN|Z1}
:ZOOM:Z2Target?
```

Example
```
:ZOOM:Z2TARGET MAIN
:ZOOM:Z2TARGET? -> :ZOOM:Z2TARGET MAIN
```

5.38 Common Command Group

The commands in this group are defined in IEEE 488.2-1987 and are independent from the instrument's individual functions. There are no front panel keys that correspond to the commands in this group.

***CAL?**

Function Executes calibration and queries the result.
Syntax `*CAL?`
Example `*CAL? -> 0`
Description This command returns 0 if calibration is completed successfully and 1 otherwise.

***CLS**

Function Clears the standard event register, extended event register, and error queue.
Syntax `*CLS`
Example `*CLS`
Description
- If the *CLS command is located immediately after the program message terminator, the output queue is also cleared.
- For information about each register and queue, see chapter 6.

***ESE**

Function Sets or queries the standard event enable register.
Syntax `*ESE {<NRf>}`
`*ESE?`
`<NRf> = 0 to 255`
Example `*ESE 251`
`*ESE? -> 251`
Description
- Specify the value as a sum of the values of each bit in decimal format.
- For example, specifying *ESE 251 will cause the standard enable register to be set to 11111011. In this case, bit 2 of the standard event register is disabled. This means that bit 5 (ESB) of the status byte register is not set to 1, even if a query error occurs.
- The default value is *ESE 0 (all bits disabled).
- A query using *ESE? will not clear the contents of the standard event enable register.
- For information about the standard event enable register, see page 6-3.

***ESR?**

Function Queries and clears the standard event register.
Syntax `*ESR?`
Example `*ESR? -> 32`
Description
- This command returns a sum of the values of each bit in decimal format.
- When an SRQ is sent, you can check what types of events have occurred.
- For example, if a value of 32 is returned, this indicates that the standard event register is set to 00100000. This means that the SRQ occurred due to a command syntax error.
- A query using *ESR? will clear the contents of the standard event register.
- For information about the standard event register, see page 6-3.

***IDN?**

Function Queries the DL850/DL850V model.
Syntax `*IDN?`
Example `*IDN? -> YOKOGAWA,DL850,0,F0.10`
Description This command returns a string in the following format: <Manufacturer>, <Model>, <Serial no.>, <Firmware version>.

***OPC**

Function Sets bit 0 (the OPC bit) of the standard event register to 1 upon the completion of the specified overlap command.
Syntax `*OPC`
Example `*OPC`
Description
- For details on how to use the *OPC command to synchronize the DL850/DL850V, see page 4-8.
- The COMMunicate:OPSE command is used to specify the overlap command.
- If *OPC is not the last command of the message, its operation is not guaranteed.

***OPC?**

Function Returns ASCII code 1 when the specified overlap command is completed.
Syntax `*OPC?`
Example `*OPC? -> 1`
Description
- For details on how to use the *OPC? command to synchronize the DL850/DL850V, see page 4-8.
- The COMMunicate:OPSE command is used to specify the overlap command.
- If *OPC? is not the last command of the message, its operation is not guaranteed.

*OPT?

Function Queries the installed options.

Syntax `*OPT?`

Example `*OPT? -> 250MW,HD,PROBEPOWER`

Description
- This command returns the <size of waveform memory> and whether the <printer>, <hard disk (interface)>, <GPIB>, <IRIG>, <user-defined computation>, and <probe power supply output> options are installed.
- Size of waveform memory: The total size of waveform data memory. “250MW” when the total size is 250 Mpoint.
  “1GW” when the total size is 1 Gpoint. “2GW” when the total size is 2 Gpoint.
  - Printer: “PRINTER”
  - Hard disk (interface): “HD” or “EHD”
  - GPIB: “GPIB”
  - IRIG: “IRIG”
  - User-defined computations: “USERDEFINE”
  - Probe power supply output: “PROBEPOWER”

  An error occurs if there is a query after this command.

*RST

Function Initializes settings.

Syntax `*RST`

Example `*RST`

Description This command also clears the *OPC and *OPC? commands that have been sent.

*SRE

Function Sets or queries the service request enable register value.

Syntax `*SRE <NRf>`
`*SRE?`
`<NRf> = 0 to 255`

Example `*SRE 239`
`*SRE? -> 175`

*STB?

Function Queries the status byte register value.

Syntax `*STB?`

Example `*STB? -> 4`

Description
- This command returns a sum of the values of each bit in decimal format.
- Because the register is read without executing serial polling, bit 6 is an MSS bit, not an RQS bit.
- For example, if a value of 4 is returned, this indicates that the status byte register is set to 00000100. This means that the error queue is not empty (in other words, an error occurred).
- A query using *STB? will not clear the contents of the status byte register.
- For information about the status byte register, see page 6-2.

*TST?

Function Performs a self-test and queries the result. The self-test consists of tests of each kind of internal memory.

Syntax `*TST?`

Example `*TST? -> 0`

Description This command returns 0 if the self-test is successful and 1 otherwise.

*WAI

Function Holds the execution of the subsequent command until the specified overlap command is completed.

Syntax `*WAI`

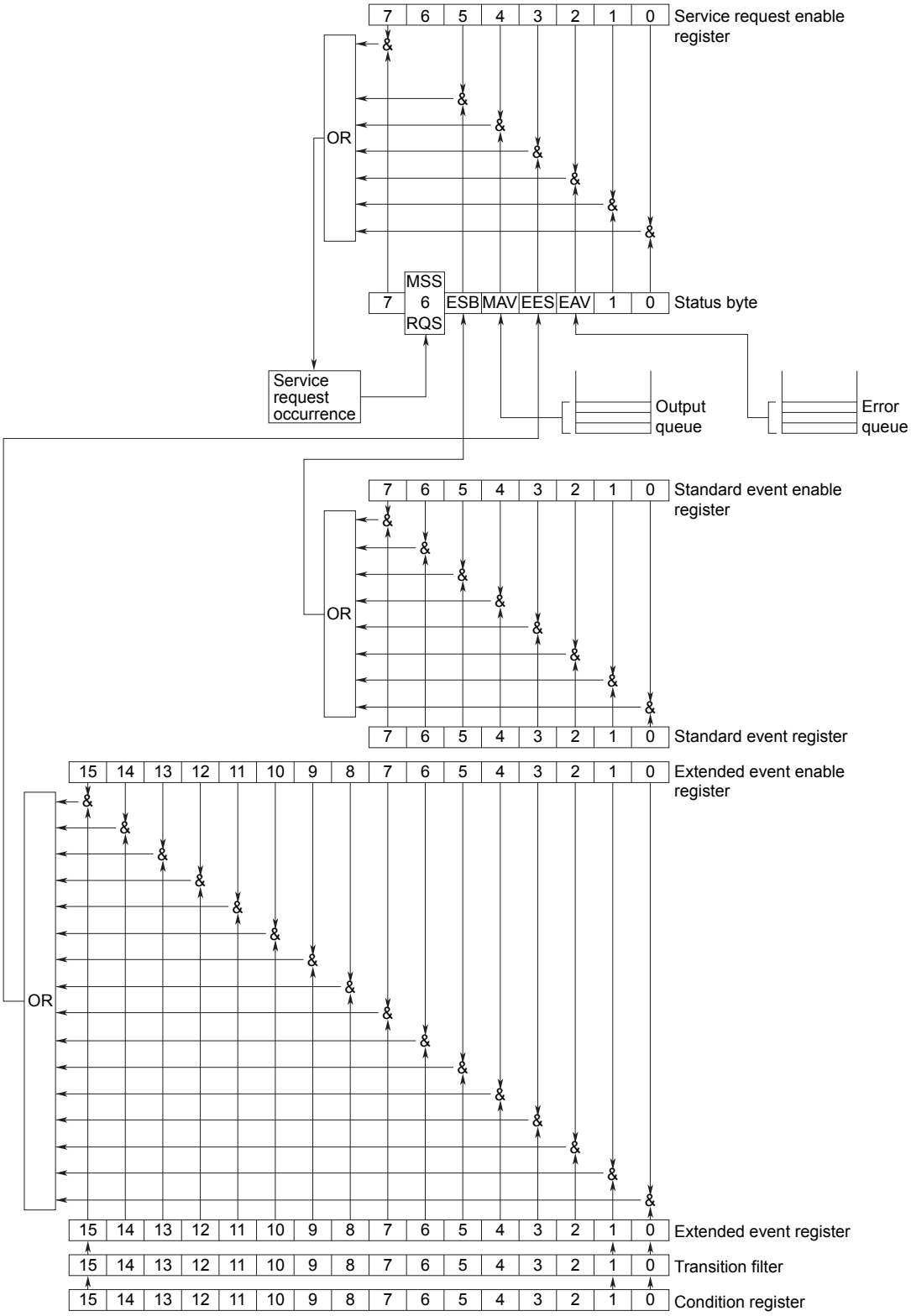
Example `*WAI`

Description
- For details on how to use the *WAI command to synchronize the DL850/DL850V, see page 4-8.
- The COMMunicate:OPSE command is used to specify the overlap command.

6.1 About Status Reports

Status Reports

The figure below shows the format of status reports that are read by serial polling. This status report format is an extended version of the status report format defined in IEEE 488.2-1992.



Overview of Registers and Queues

Name	Function	Write	Read
Status byte	–	–	Serial polling `(RQS)`, `*STB? (MSS)`
Service request enable register	Status byte mask	`*SRE`	`*SRE?`
Standard event register	Indicates device status changes	–	`*ESR?`
Standard event enable register	Standard event register mask	`*ESE`	`*ESE?`
Extended event register	Indicates device status changes	–	`STATus:EESR?`
Extended event enable register	Extended event register mask	`STATus:EESE`	`STATus:EESE?`
Condition register	Current device status	–	`STATus:CONDition?`
Transition filter	Conditions that change the extended event register	`STATus:FILTer<x>`	`STATus:FILTer<x>?`
Output queue	Stores response messages for queries	Query commands	
Error Queue	Stores error numbers and messages	–	`STATus:ERRor?`

Registers and Queues That Affect the Status Byte

The following registers affect the status byte bits.

Standard event register	Sets bit 5 (ESB) of the status byte to 1 or 0.
Output queue	Sets bit 4 (MAV) of the status byte to 1 or 0.
Extended event register	Sets bit 3 (EES) of the status byte to 1 or 0.
Error queue	Sets bit 2 (EAV) of the status byte to 1 or 0.

Enable Registers

The following registers are used to mask a bit so that the bit will not affect the status byte even when it is set to 1.

Service request enable register	Masks bits of the status byte.
Standard event register	Masks bits in the standard event register.
Extended event register	Masks bits in the extended event register.

Reading and Writing to Registers

For example, you can use the `*ESE` command to set the standard event register bits to ones and zeros. You can use the `*ESE?` command to query whether the standard event register bits are ones or zeros. For details on commands, see chapter 5.

6.2 Status Byte

Status Byte

- **Bits 0 and 7**
  Not used (always zero)
- **Bit 1**
  Reserved
- **Bit 2 EAV (Error Available)**
  This bit is 1 when the error queue is not empty. In other words, this bit is set to 1 when an error occurs. For details, see page 6-6.
- **Bit 3 EES (Extend Event Summary Bit)**
  This bit is 1 when the logical AND of the extended event register and its corresponding event register is 1. In other words, this bit is set to 1 when a certain event takes place inside the instrument. For details, see page 6-5.
- **Bit 4 MAV (Message Available)**
  This bit is 1 when the output queue is not empty. In other words, this bit is set to 1 when there is data to be transmitted in response to a query. For details, see page 6-6.
- **Bit 5 ESB (Event Summary Bit)**
  This bit is 1 when the logical AND of the standard event register and its corresponding event register is 1. In other words, this bit is set to 1 when a certain event takes place inside the instrument. For details, see page 6-4.
- **Bit 6 RQS (Request Service)/MSS (Master Status Summary)**
  This bit is 1 when the logical AND of the status byte excluding bit 6 and the service request enable register is 1. In other words, this bit is set to 1 when the instrument requests service from the controller. RQS is set to 1 when the MSS bit changes from 0 to 1 and is cleared when serial polling is carried out or when the MSS bit changes to 0.

Bit Masking

To mask a bit in the status byte so that it does not trigger an SRQ, set the corresponding bit of the service request enable register to zero.
For example, to mask bit 2 (EAV) so that service is not requested when an error occurs, set bit 2 of the service request enable register to 0. Do this using the `*SRE` command. To query whether each bit of the service request enable register is 1 or 0, use `*SRE?`. For details on the `*SRE` command, see chapter 5.

Status Byte Operation

A service request is issued when bit 6 in the status byte becomes 1. Bit 6 is set to 1 when any other bit becomes 1 (when the corresponding bit of the service request enable register is also set to 1). For example, if an event occurs and the logical AND of a standard event register bit and its corresponding enable register bit is 1, then bit 5 (ESB) is set to 1. At this point, if bit 5 of the service request enable register is 1, bit 6 (MSS) is set to 1, and the DL850/DL850V requests service from the controller.
You can check what type of event occurred by reading the contents of the status byte.

Reading the Status Byte

There are two ways to read the contents of the status byte.

- ***STB? Query**
  Bit 6 functions as MSS when a query is made using `*STB?`. This causes the MSS to be read. This query does not cause any of the status byte bits to be cleared after the status byte is read.
- **Serial Polling**
  Serial polling causes bit 6 to function as an RQS bit. This causes the RQS to be read. After the status byte is read, only the RQS bit is cleared. You cannot read the MSS bit when serial polling is used.

Clearing the Status Byte

There is no way to clear all of the bits in the status byte. The bits that are cleared vary for each operation as follows:

- ***STB? Query**
  None of the bits are cleared.
- **Serial Polling**
  Only the RQS bit is cleared.
- **A *CLS command is received.**
  When a `*CLS` command is received, the status byte itself is not cleared, but the contents of the standard event register, which affect the bits in the status byte, are cleared. As a result, the corresponding status byte bits are cleared. Because the output queue is not cleared with a `*CLS` command, bit 4 (MAV) in the status byte is not affected. However, the output queue will be cleared if the `*CLS` command is received just after a program message terminator.

6.3 Standard Event Register

Standard Event Register

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

- **Bit 7 PON (Power ON)**
  This bit is set to 1 when the DL850/DL850V is turned on.
- **Bit 6 URQ (User Request)**
  Not used (always zero)
- **Bit 5 CME (Command Error)**
  This bit is set to 1 when there is a command syntax error.
  Example Incorrectly spelled command name; 9 used in octal data.
- **Bit 4 EXE (Execution Error)**
  This bit is set to 1 when the command syntax is correct, but the command cannot be executed in the current state.
  Example The DL850/DL850V receives a command whose parameter is outside the selectable range. An attempt is made to print a hard copy while the DL850/DL850V is running.
- **Bit 3 DDE (Device Error)**
  This bit is set to 1 when a command cannot be executed for internal reasons other than a command syntax error or command execution error.
- **Bit 2 QYE (Query Error)**
  This bit is set to 1 when a query command is received, but the output queue is empty or the data is lost.
  Example There is no response data. Data is lost due to an overflow in the output queue.
- **Bit 1 RQC (Request Control)**
  Not used (always zero)
- **Bit 0 OPC (Operation Complete)**
  This bit is set to 1 upon the completion of the operation designated by the `*OPC` command (see chapter 5).

Bit Masking

To mask a certain bit of the standard event register so that it does not cause bit 5 (ESB) in the status byte to change, set the corresponding bit of the standard event enable register to zero.
For example, to mask bit 2 (QYE) so that ESB will not be set to 1 even if a query error occurs, set bit 2 of the standard event enable register to zero. Do this using the `*ESE` command. To query whether each bit of the standard event enable register is 1 or 0, use `*ESE?`.
For details on the `*ESE` command, see chapter 5.

Standard Event Register Operation

The standard event register indicates eight types of events that occur inside the DL850/DL850V. When one of the bits in this register becomes 1 (and the corresponding bit of the standard event enable register is also 1), bit 5 (ESB) in the status byte is set to 1.
Example
1. A query error occurs.
2. Bit 2 (QYE) is set to 1.
3. If bit 2 of the standard event enable register is 1, bit 5 (ESB) in the status byte is set to 1.

You can also check what type of event occurred in the DL850/DL850V by reading the contents of the standard event register.

Reading the Standard Event Register

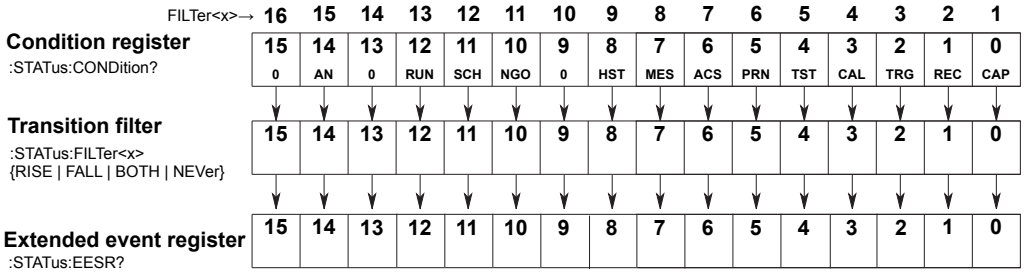
You can use the `*ESR?` command to read the contents of the standard event register. The register is cleared after it is read.

Clearing the Standard Event Register

The standard event register is cleared when:
- The contents of the standard event register are read using the `*ESR?` command.
- A `*CLS` command is received.
- The DL850/DL850V is turned off and then back on.

6.4 Extended Event Register

The extended event register receives information about changes in the condition register, which indicates the DL850/DL850V's internal condition. The information is the result of edge detection performed by the transition filter.


FILTer<x>→ 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
Condition register
:STATus:CONDition?
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
0 AN 0 RUN SCH NGO 0 HST MES ACS PRN TST CAL TRG REC CAP
Transition filter
:STATus:FILTer<x>
{RISE | FALL | BOTH | NEVer}
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Extended event register
:STATus:EESR?
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

The condition register bits are described below.

Bit 0	CAP (Capture)	This bit is 1 when waveform acquisition is in progress.
Bit 1	REC (Record)	This bit is 1 when recording is in progress.
Bit 2	TRG (Awaiting Trigger)	This bit is 1 when the DL850/DL850V is waiting for a trigger.
Bit 3	CAL (Calibration)	This bit is 1 when calibration is in progress.
Bit 4	TST (Testing)	This bit is 1 when a self-test is in progress.
Bit 5	PRN (Printing)	This bit is 1 when the internal printer is operating or when data is being transmitted to a network printer.
Bit 6	ACS (Accessing)	This bit is 1 when a drive is being accessed.
Bit 7	MES (Measuring)	This bit is 1 when automated measurement of waveform parameters is in progress.
Bit 8	HST (History Search)	This bit is 1 when a history search is in progress.
Bit 10	NGO (Go/No-Go)	This bit is 1 when a GO/NO-GO search is in progress.
Bit 11	SCH (Search)	This bit is 1 when a search is in progress.
Bit 12	RUN (Running)	This bit is 1 when a measurement is in progress.
Bit 14	AN (Analysis)	This bit is 1 when an analysis is in progress.

The transition filter parameters detect changes in the specified condition register bits (numeric suffixes 1 to 16) and overwrite the extended event register in the following ways.

RISE	The specified extended event register bit is set to 1 when the corresponding condition register bit changes from 0 to 1.
FALL	The specified extended event register bit is set to 1 when the corresponding condition register bit changes from 1 to 0.
BOTH	The specified extended event register bit is set to 1 when the corresponding condition register bit changes from 0 to 1 or from 1 to 0.
NEVer	Always zero.

6.5 Output and Error Queues

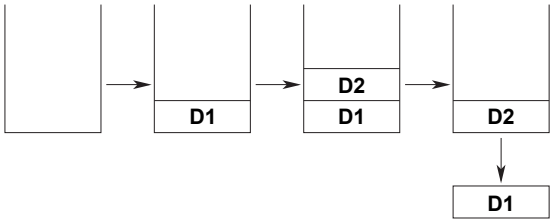
Output Queue

The output queue stores query response messages. For example, if you send a `:WAVeform:SEND?` command, which requests for the transmission of acquired data, the data is stored in the output queue until it is read.

As shown below, data is stored in order and read from the oldest message first. The output queue is cleared when:

- A new message is received from the controller.
- A deadlock occurs (see page 4-2).
- A device clear command (DCL or SDC) is received.
- The DL850/DL850V is turned off and then back on.

The `*CLS` command does not clear the output queue. You can determine whether the output queue is empty by checking bit 4 in the status byte (MAV).



Error Queue

When an error occurs, the error queue stores the error number and message. For example, if the DL850/DL850V receives an incorrect program message from the controller, the error number (113) and the error message (“Undefined header”) are stored in the error queue when the DL850/DL850V displays the error message.

You can use the `:STATus:ERRor?` query to read the contents of the error queue. Like the output queue, the messages in the error queue are read from the oldest one first.

If the error queue overflows, the last message is replaced with the following message: 350, “Queue overflow.”

The error queue is cleared when:

- A `*CLS` command is received.
- The DL850/DL850V is turned off and then back on.

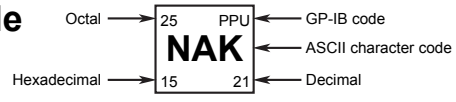
You can determine whether the error queue is empty by checking bit 2 in the status byte (EAV).

Appendix 1 ASCII Character Codes

The following table contains ASCII character codes.

	0	1	2	3	4	5	6	7
0	0 **NUL** 0 0	20 **DEL** 10 16	40 0 **SP** 20 32	60 16 **0** 30 48	100 0 **@** 40 64	120 16 **P** 50 80	140 0 **‘** 60 96	160 16 **p** 70 112
1	1 GTL **SOH** 1 1	21 LLO **DC1** 11 17	41 1 **!** 21 33	61 17 **1** 31 49	101 1 **A** 41 65	121 17 **Q** 51 81	141 1 **a** 61 97	161 17 **q** 71 113
2	2 **STX** 2 2	22 **DC2** 12 18	42 2 **”** 22 34	62 18 **2** 32 50	102 2 **B** 42 66	122 18 **R** 52 82	142 2 **b** 62 98	162 18 **r** 72 114
3	3 **ETX** 3 3	23 **DC3** 13 19	43 3 **#** 23 35	63 19 **3** 33 51	103 3 **C** 43 67	123 19 **S** 53 83	143 3 **c** 63 99	163 19 **s** 73 115
4	4 SDC **EOT** 4 4	24 DCL **DC4** 14 20	44 4 **$** 24 36	64 20 **4** 34 52	104 4 **D** 44 68	124 20 **T** 54 84	144 4 **d** 64 100	164 20 **t** 74 116
5	5 PPC **ENQ** 5 5	25 PPU **NAK** 15 21	45 5 **%** 25 37	65 21 **5** 35 53	105 5 **E** 45 69	125 21 **U** 55 85	145 5 **e** 65 101	165 21 **u** 75 117
6	6 **ACK** 6 6	26 **SYN** 16 22	46 6 **&** 26 38	66 22 **6** 36 54	106 6 **F** 46 70	126 22 **V** 56 86	146 6 **f** 66 102	166 22 **v** 76 118
7	7 **BEL** 7 7	27 **ETB** 17 23	47 7 **’** 27 39	67 23 **7** 37 55	107 7 **G** 47 71	127 23 **W** 57 87	147 7 **g** 67 103	167 23 **w** 77 119
8	10 GET **BS** 8 8	30 SPE **CAN** 18 24	50 8 **(** 28 40	70 24 **8** 38 56	110 8 **H** 48 72	130 24 **X** 58 88	150 8 **h** 68 104	170 24 **x** 78 120
9	11 TCT **HT** 9 9	31 SPD **EM** 19 25	51 9 **)** 29 41	71 25 **9** 39 57	111 9 **I** 49 73	131 25 **Y** 59 89	151 9 **i** 69 105	171 25 **y** 79 121
A	12 **LF** A 10	32 **SUB** 1A 26	52 10 ***** 2A 42	72 26 **:** 3A 58	112 10 **J** 4A 74	132 26 **Z** 5A 90	152 10 **j** 6A 106	172 26 **z** 7A 122
B	13 **VT** B 11	33 **ESC** 1B 27	53 11 **+** 2B 43	73 27 **;** 3B 59	113 11 **K** 4B 75	133 27 **[** 5B 91	153 11 **k** 6B 107	173 27 **{** 7B 123
C	14 **FF** C 12	34 **FS** 1C 28	54 12 **,** 2C 44	74 28 **<** 3C 60	114 12 **L** 4C 76	134 28 **** 5C 92	154 12 **l** 6C 108	174 28 **|** 7C 124
D	15 **CR** D 13	35 **GS** 1D 29	55 13 **-** 2D 45	75 29 **=** 3D 61	115 13 **M** 4D 77	135 29 **]** 5D 93	155 13 **m** 6D 109	175 29 **}** 7D 125
E	16 **SO** E 14	36 **RS** 1E 30	56 14 **.** 2E 46	76 30 **>** 3E 62	116 14 **N** 4E 78	136 30 **^** 5E 94	156 14 **n** 6E 110	176 30 **~** 7E 126
F	17 **SI** F 15	37 **US** 1F 31	57 15 **/** 2F 47	77 UNL **?** 3F 63	117 15 **O** 4F 79	137 UNT **_** 5F 95	157 15 **o** 6F 111	177 **DEL (RUBOUT)** 7F 127
	Address command	Universal command	Listener address		Talker address		Secondary command	

Example



Appendix 2 Error Messages

This section explains communication error messages.

- Messages can be displayed in English or Japanese on the DL850/DL850V. However, when they are read from a PC or other similar device, messages are displayed in English.
- If servicing is necessary to solve the problem indicated by a message, contact your nearest YOKOGAWA dealer.
- Only communication error messages are listed here. For details on other error messages, see the user's manual, IM DL850-02EN.
  - Communication syntax errors 100 to 199
  - Communication execution errors 200 to 299
  - Device-specific and other errors 300 to 399
  - Communication query errors 400 to 499
  - System errors (communication) 300 and 399

  } Listed below

Communication Syntax Errors (100 to 199)

Code	Message	Corrective Action	Page
102	Syntax error.	A syntax error not covered by error codes 100 to 199.	Chapters 4 and 5
103	Invalid separator.	Separate data values with a comma.	4-1
104	Data type error.	See page 4-6 and 4-7 and use the correct data type for each parameter.	4-6 and 4-7
105	GET not allowed.	GET is not supported as a response to an interface message.	3-8
108	Parameter not allowed.	Check the number of data values.	4-6 and chapter 5
109	Missing parameter.	Be sure to include all necessary data values.	4-6 and chapter 5
111	Header separator error.	Use a comma to separate each header from its data.	4-1
112	Program mnemonic too long.	Check the command length.	Chapter 5
113	Undefined header.	Check the header.	Chapter 5
114	Header suffix out of range.	Check the header.	Chapter 5
120	Numeric data error.	A value must be specified where the syntax contains <NRf>.	4-6
123	Exponent too large.	Where the syntax contains <NR3>, make the exponent that follows E smaller.	4-6 and chapter 5
124	Too many digits.	Limit numeric values to 255 digits or less.	4-6 and chapter 5
128	Numeric data not allowed.	Use a data type other than <NRf>.	4-6 and chapter 5
131	Invalid suffix.	Check the units where the syntax contains <Voltage>, <Time>, or <Frequency>.	4-6
134	Suffix too long.	Check the units where the syntax contains <Voltage>, <Time>, or <Frequency>.	4-6
138	Suffix not allowed.	Units of measurement can only be used where the syntax contains <Voltage>, <Time>, or <Frequency>.	4-6
141	Invalid character data.	Be sure to select one of the listed choices when the syntax contains {...|...|...}.	Chapters 4 and 5
144	Character data too long.	Check the spelling of the strings when the syntax contains {...|...|...}.	Chapter 5
148	Character data not allowed.	Use a data type other than <String>.	Chapter 5
150	String data error.	Enclose parameters with single or double quotation marks where the syntax contains <String>.	4-7
151	Invalid string data.	The parameter is either too long, or it contains an unusable character.	Chapter 5
158	String data not allowed.	Use a data type other than <String>.	Chapter 5
161	Invalid block data.	<Block data> cannot be used.	4-7 and chapter 5
168	Block data not allowed.	<Block data> cannot be used.	4-7 and chapter 5

Code	Message	Corrective Action	Page
171	Missing Right	Mathematical operations cannot be used.	—
172	Invalid expression.	Mathematical operations cannot be used.	Chapter 5
178	Expression data not allowed.	Mathematical operations cannot be used.	Chapter 5
181	Invalid outside macro definition.	The DL850/DL850V does not support the IEEE 488.2 macro specifications.	—

Communication Execution Errors (200 to 299)

Code	Message	Corrective Action	Page
221	Setting conflict.	Check settings that are related to each other.	Chapter 5
222	Data out of range.	Check the ranges of the settings.	Chapter 5
223	Too much data.	Check data byte lengths.	Chapter 5
224	Illegal parameter value.	Check the ranges of the settings.	Chapter 5
225	OverFlow.	Keep program messages to 1024 bytes or less in length, including <PMT>.	4-2
226	Out Of Memory.	Keep program messages to 1024 bytes or less in length, including <PMT>.	4-2
241	Hardware missing.	Check that the specified options are all installed.	—
260	Expression error.	Mathematical operations cannot be used.	—
270	Macro error.	The DL850/DL850V does not support the IEEE 488.2 macro specifications.	—
272	Macro execution error.	The DL850/DL850V does not support the IEEE 488.2 macro specifications.	—
273	Illegal macro label.	The DL850/DL850V does not support the IEEE 488.2 macro specifications.	—
275	Macro definition too long.	The DL850/DL850V does not support the IEEE 488.2 macro specifications.	—
276	Macro recursion error.	The DL850/DL850V does not support the IEEE 488.2 macro specifications.	—
277	Macro redefinition not allowed.	The DL850/DL850V does not support the IEEE 488.2 macro specifications.	—
278	Macro header not found.	The DL850/DL850V does not support the IEEE 488.2 macro specifications.	—

Communication Query Errors (400 to 499)

Code	Message	Corrective Action	Page
410	Query INTERRUPTED.	Check the transmission and reception order.	4-2
420	Query UNTERMINATED.	Check the transmission and reception order.	4-2
430	Query DEADLOCKED.	Keep program messages to 1024 bytes or less in length, including <PMT>.	4-2
440	Query UNTERMINATED after indefinite response.	Do not write a query after *IDN? or *OPT?.	—

System Communication Errors (300 and 399)

Code	Message	Corrective Action	Page
300	Communication device-specific error.	Servicing required.	—
399	Fatal error in the communication driver.	Servicing required.	—

Other Error (350)

Code	Message	Corrective Action	Page
350	Queue overflow.	Read the error queue.	6-6

Note

Code 350 occurs when the error queue overflows. This error is only returned in response to a `:STATus:ERRor?` query; it is never displayed on the screen.

Appendix 3 About the IEEE 488.2-1992 Standard

The DL850/DL850V's GP-IB interface conforms to the IEEE 488.2-1992 standard. This standard specifies that the following 23 items be stated in the document. This section describes these items.

(1) Of the IEEE 488.1 interface functions, the subsets that are supported
See section 3.4, "GP-IB Interface Specifications."

(2) The operation of the device when it is assigned an address outside the 0 to 30 range
The address of this instrument cannot be set to an address outside the 0 to 30 range.

(3) Reaction of the device when the user changes the address
The address change is detected when the user presses UTILITY and then the Remote Ctrl soft key, and changes the address. The new address is valid until the next time it is changed.

(4) Device settings at power-on. The commands that can be used at power-on.
As a basic rule, the previous settings (the settings that were in use when the DL850/DL850V was turned off) are used.
There are no limitations on the commands that can be used at power-on.

(5) Message exchange options
(a) Input buffer size
4096 bytes.

(b) Queries that return multiple response messages
See the example of the commands given in chapter 5.

(c) Queries that create response data when the command syntax is being analyzed
All queries create response data when the command syntax is analyzed.

(d) Queries that create response data during reception
There are no queries of which the response data are created upon receiving a send request from the controller.

(e) Commands that have parameters that restrict one another
There are commands such as :CHANnel<x>:FREQ:INOut:PROBe and CHANnel<x>:FREQ:VDIV that have parameters that place restrictions unilaterally, but there are no commands that have parameters that restrict one another.

(6) Items that are included in the functional or composite header elements constituting a command
See chapters 4 and 5.

(7) Buffer sizes that affect block data transmission
When block data is being transmitted, the output queue is expanded to match the size of the data that is being transmitted.

(8) A list of program data elements that can be used in equations and their nesting limitations
Equations cannot be used.

(9) Syntax of the responses to queries
See the example of the commands given in chapter 5.

(10) Communication between devices that do not follow the response syntax is not supported by the DL850/DL850V.

(11) Size of the response data block
1 to 2004000 bytes

(12) A list of supported common commands
See section 5.38, "Common Command Group."

(13) Device condition after a successful calibration
The device will be performing measurements.

(14) The maximum length of block data that can be used for the *DDT trigger macro definition
Not supported.

(15) The maximum length of the macro label for defining macros, the maximum length of block data that can be used for the macro definition, and the process when recursion is used in macro definitions
Macro functions are not supported.

(16) Reply to the *IDN? query
See section 5.38, "Common Command Group."

(17) Size of storage area for protected user data for PUD and *PUD?
*PUD and *PUD? are not supported.

(18) The length of the *RDT and *RDT? resource names
*RDT and *RDT? are not supported.

(19) The change in the status due to *RST, *LRN?, *RCL, and *SAV

***RST and *LRN?**

See section 5.38, “Common Command Group.”

***RCL and *SAV**

These common commands are not supported.

(20) The extent of the self-test using the *TST? command

Performs the same internal memory test that is executed when the user presses UTILITY and then the Self Test soft key, and executes the MEMORY test.

(21) The structure of the extended return status

See chapter 6.

(22) Whether each command is processed in an overlapped manner or sequentially

See section 4.5, “Synchronization with the Controller” and chapter 5.

(23) The description of the execution of each command

See the explanations of each command’s function in chapter 5; the features guide, IM DL850-01EN; and the user’s manual, IM DL850-02EN.

Index

A

Page

abbreviated form ... 4-5
analysis-source start point (FFT) ... 5-74
analysis source points, number (FFT) ... 5-74
angle cursor (T-Y) ... 5-60
ASCII character codes ... App-1
attenuation constant (FFT) ... 5-74
attenuation constant (MATH) ... 5-92
auto naming (capture) ... 5-29
auto naming (file) ... 5-78
auto naming (hard disk recording) ... 5-26
auto scrolling ... 5-139
auto setup ... 5-27
average, number of times (frequency) ... 5-41
average count (FFT) ... 5-74
average count (MATH) ... 5-92
averaging domain (FFT) ... 5-74
averaging domain (MATH) ... 5-92

B

Page

background ... 5-89
backlight, automatic off ... 5-119
balancing (strain) ... 5-44
bandwidth limit (16CH voltage) ... 5-51
bandwidth limit (acceleration) ... 5-37
bandwidth limit (temperature) ... 5-47
beep (capture) ... 5-29
beep (GO/NO-GO) ... 5-80
beep (trigger) ... 5-122
block data ... 4-7
Boolean ... 4-7
bridge voltage (strain) ... 5-45
burnout (temperature) ... 5-47

C

Page

calibration ... 5-28, 5-141
capture window ... 5-30
center frequency (frequency) ... 5-40
character data ... 4-7
character notations ... iv
chattering elimination (acceleration) ... 5-37
chattering elimination (logic) ... 5-44
clear trace ... 5-55
click sound ... 5-117
commands ... 4-3
comment (file) ... 5-79
comment (hard disk recording) ... 5-26
comment (screen capture) ... 5-89
comment (screen printing) ... 5-85
common command header ... 4-3
computation range ... 5-95
computation window (FFT) ... 5-74
condition register ... 5-114
current ... 4-6
cutoff frequency ... 5-94
cycle averaging ... 5-92
cycle trace (measure) ... 5-103
cyclic statistical processing (measure) ... 5-102

D

Page

damping rate (FFT) ... 5-74
damping rate (MATH) ... 5-94
data ... 4-6
data loading ... 5-78
data saving ... 5-78
data saving (capture) ... 5-29
data saving (screen capture) ... 5-89
data saving (trigger) ... 5-123
data transfer rate (Ethernet) ... 1-2
data transfer rate (GP-IB) ... 3-5
data transfer rate (USB) ... 2-2
date ... 5-117
DCL (Device Clear) ... 3-7
deadlock ... 4-2
deceleration prediction (frequency) ... 5-40
delay ... 5-97
delay (trigger) ... 5-126
determination level (search) ... 5-108
determination logic ... 5-82
determination mode ... 5-82
determination range (GO/NO-GO) ... 5-83
determination range (history) ... 5-87
determinations, number of ... 5-82
distal ... 5-99
distance per pulse (frequency) ... 5-40
dual capture ... 5-29
dual capture length ... 5-30

E

Page

e-mail transmission (capture) ... 5-29
e-mail transmission (GO/NO-GO) ... 5-81
e-mail transmission (trigger) ... 5-122
edge polarity (search) ... 5-108
error messages ... App-2
error queue ... 5-114, 6-6
Ethernet interface ... 1-2
exponential averaging, attenuation constant ... 5-25
extended event register ... 6-5

F

Page

FFT cursor (marker cursor) ... 5-58
FFT cursor (peak cursor) ... 5-59
FFT points (MATH) ... 5-94
file copying ... 5-77
file deletion ... 5-77
filename ... 4-7
file name (hard disk recording) ... 5-26
file saving ... 5-78
filter (frequency) ... 5-40
filter (MATH) ... 5-94
filter (strain) ... 5-45
filter band ... 5-94
filter type ... 5-94
font size ... 5-120
format (capture) ... 5-30
format (monitor) ... 5-104
format (system) ... 5-120
format (waveform) ... 5-134
format (zoom) ... 5-139
frequency ... 4-6
front panel ... 1-1, 2-1, 3-1

G

gain (acceleration) ... 5-32
gain (monitor) ... 5-104
gauge factor (strain) ... 5-45
GP-IB board ... 3-2
GP-IB interface ... 3-4, 3-5
groups ... 4-3
GTL (Go To Local) ... 3-7

H

hard disk recording ... 5-25, 5-26
H cursor (X-Y) ... 5-66
hold-off time ... 5-127
horizontal cursor (T-Y) ... 5-61
horizontal magnification (zoom) ... 5-140
horizontal position (capture) ... 5-31
hysteresis (frequency) ... 5-38

I

IFC (Interface Clear) ... 3-7
Initialization ... 5-90
input coupling (16CH voltage) ... 5-51
input coupling (acceleration/voltage) ... 5-32
input coupling (analog voltage) ... 5-48
input coupling (frequency) ... 5-38
input coupling (temperature) ... 5-47
intensity ... 5-71
IRIG format ... 5-118
IRIG modulation ... 5-118

K

key lock ... 5-118

L

LCD brightness ... 5-119
linear scaling (16CH voltage) ... 5-52
linear scaling (analog voltage) ... 5-48
linear scaling (frequency) ... 5-39
linear scaling (strain) ... 5-45
LLO (Local Lockout) ... 3-7
local lockout ... 5-56
log, starting ... 5-91
logic (history) ... 5-87
logic condition (history) ... 5-88

M

manual trigger ... 5-106
marker cursor (T-Y) ... 5-62
marker cursor (X-Y) ... 5-66
measurement pulse (frequency) ... 5-42
medium, free space ... 5-77
menu language ... 5-120
mesial ... 5-99
message language ... 5-119
messages ... 4-1
moving average order (frequency) ... 5-41

N

no-go count ... 5-82

O

offset (16CH voltage) ... 5-52, 5-54
offset (analog voltage) ... 5-49, 5-50
offset (frequency) ... 5-40
offset (monitor) ... 5-104
one cycle mode ... 5-103
output queue ... 6-6
over-limit reset (frequency) ... 5-42
overlap commands ... 4-8

P

P1X|P2X (16CH voltage) ... 5-53
P1X|P2X (analog voltage) ... 5-49
P1X|P2X (frequency) ... 5-39
P1X|P2X (strain) ... 5-46
pen marker ... 5-137
phase shift (external clock) ... 5-93
phase shift (internal clock) ... 5-95
preset (frequency) ... 5-38
probe (analog voltage) ... 5-50
probe attenuation (frequency) ... 5-38
program message ... 4-1
proximal ... 5-99
pull-up (frequency) ... 5-38
pulse (frequency) ... 5-42
pulse/rotate ... 5-25
pulse averaging (frequency) ... 5-41
pulse count reset (frequency) ... 5-42
pulses per rotation (frequency) ... 5-42

Q

query ... 4-1

R

range ... 5-105
record length ... 5-25
reference condition (history) ... 5-86
reference condition (parameter) ... 5-82
reference condition (waveform zone) ... 5-83
register ... 4-7
remote and local modes, switching ... 1-2, 2-2, 3-4
remote or local ... 5-56
remote terminal ... 5-120
REN (Remote Enable) ... 3-7
response ... 4-5
response message ... 4-1
responses without headers ... 4-5
RJC (temperature) ... 5-47

S

sample rate ... 5-121
save file name ... 5-79
scaling (MATH) ... 5-95
scaling coefficient (16CH voltage) ... 5-52
scaling coefficient (analog voltage) ... 5-48
scaling coefficient (frequency) ... 5-39
scaling coefficient (strain) ... 5-45
screen, base color ... 5-70
screen capture printing (GO/NO-GO) ... 5-80
screen capture printing (trigger) ... 5-122
screen capture saving (GO/NO-GO) ... 5-80
screen capture saving (trigger) ... 5-122
SDC (Selected Device Clear) ... 3-7
search (history) ... 5-86
search end position ... 5-109
search start position ... 5-109
self-test ... 5-142
sensitivity (acceleration) ... 5-33
shunt calibration (strain) ... 5-46
side panel ... 1-1, 2-1, 3-1
single start ... 5-112

slope (frequency) 5-38
smoothing (frequency) 5-41
snapshot 5-111
SNTP 5-117
SPD (Serial Poll Disable) 3-7
SPE (Serial Poll Enable) 3-7
standard event register 6-4
START/STOP key 5-118
status byte 6-3
status reports 6-1
stop prediction (frequency) 5-42
string data 4-7

T

Page
T/DIV 5-140
T/DIV (capture) 5-31
T/DIV (time base) 5-121
TCP/IP 1-5
thermocouple (temperature) 5-47
threshold level (frequency) 5-38
threshold level (MATH) 5-93
time 4-6, 5-117
time base 5-25
time base (TIME/DIV) 5-121
timeout (cursor) 5-69
timeout (GO/NO-GO) 5-84
timeout (measure) 5-103
timeout (single start) 5-112
time search, time of 5-110
time synchronization 5-120
time unit (frequency) 5-42
time window 5-94
TMCTL ii
trace-clear-on-start 5-137
transition filter 6-5
trigger
A->B(n) 5-122
action-on-trigger 5-122
A Delay B 5-123
AND 5-123
B<Time 5-125
B>Time 5-125
B Between 5-125
B TimeOut 5-125
EdgeOnA 5-126
enhanced 5-129
logic 5-129
OR 5-128
simple 5-129
T<T1,T2<T 5-131
T<Time 5-131
T>Time 5-130
T1<T<T2 5-131
time 5-131
wave window 5-130
trigger date (history) 5-87
trigger position 5-128
trigger time (history) 5-88

U

Page
unit (16CH voltage) 5-53
unit (acceleration) 5-33
unit (analog voltage) 5-50
unit(CAN) 5-36
unit (frequency) 5-39, 5-42
unit (MATH) 5-96
unit (strain) 5-46
unit (temperature) 5-48
upper-level query 4-4
USB cable 2-3
USB hub 2-3
USB interface 2-2
USB keyboard 5-120
user verification 1-3

V

Page
V/DIV (16CH voltage) 5-54
V/DIV (analog voltage) 5-51
Value/Div (frequency) 5-43
V cursor (T-Y) 5-64
V cursor (X-Y) 5-68
vertical position (16CH voltage) 5-54
vertical position (acceleration) 5-32
vertical position (analog voltage) 5-50
vertical position (frequency) 5-40
vertical position (logic) 5-44
vertical scale (16CH voltage) 5-54
vertical scale (analog voltage) 5-50
vertical scale, center value 5-75
vertical scale sensitivity (FFT) 5-75
vertical zoom factor (16CH voltage) 5-54
vertical zoom factor (acceleration) 5-33
vertical zoom factor (analog voltage) 5-51
vertical zoom factor (frequency) 5-43
vertical zoom factor (logic) 5-44
voltage 4-6
voltage range (frequency) 5-38

W

Page
waveform accumulation, number of 5-70
waveform acquisition 5-25
waveform acquisitions, number of 5-25
waveform interpolation 5-71
waveform label (frequency) 5-43
waveform label (MATH) 5-95
waveform saving (GO/NO-GO) 5-81
Window1 and Window2, combined display 5-137
window layout 5-140

Z

Page
zoom factor (capture) 5-31
zoom position 5-140